APX ATARI® PROGRAM EXCHANGE

Basic Utility Diskette

APX-20036

User-Written Software for ATARI Home Computers

Basic Utility Diskette

APX-20036

BASIC UTILITY DISKETTE

BASIC Program Compressor (MASHER)
APX-20008

BASIC Cross-reference Utility (XREF)
APX-20009

Ultimate Renumber Utility
APX-20086

Variable Changer
APX-20012

TRADEMARKS OF ATARI

The following are trademarks of Atari, Inc.

ATARI®
ATARI 400™ Home Computer
ATARI 800™ Home Computer
ATARI 410™ Program Recorder
ATARI 810™ Disk Drive
ATARI 820™ 40-Column Printer
ATARI 822™ Thermal Printer
ATARI 825™ 80-Column Printer
ATARI 830™ Acoustic Modem
ATARI 850™ Interface Module

****
Distributed by

The ATARI Program Exchange
P. O. Box 427
155 Moffett Park Drive, B-1
Sunnyvale, CA 94086

To request an APX Software Catalog, write to the address above, or call toll-free:

800/538-1862 (outside California)
800/672-1850 (within California)

Or call our Sales number, 408/745-5535.
****

PREFACE

This manual is a collection of the four manuals accompanying the programs on the BASIC UTILITY DISKETTE. They are in the following order:

1. BASIC Program Compressor (MASHER)

2. BASIC Cross-reference Utility (XREF)

3. Ultimate Renumber Utility

4. Variable Changer

Each manual is reproduced in its entirety, including separate Review Forms, so that you can send us your comments about each program and manual as you use them.

BASIC PROGRAM COMPRESSOR
(MASHER)

by

Dale Yocum
(modified by Jerry White)

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 ultility program to help you squeeze every last byte out of your 8K BASIC software. It compresses your BASIC programs into as litle 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 unknowlingy 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 <u>long</u> 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!

NAM[illegible] [illegible] SUCH AS
① [illegible]
[illegible] PROG[illegible]
[illegible] LIKE RETURN AS
[illegible] WORD IN
A [illegible] "
[illegible] RETURN KEY
LL 320 REM "[illegible] RETURN KEY"

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.

LIMITED WARRANTY ON MEDIA AND HARDWARE ACCESSORIES.

We, Atari, Inc., guarantee to you, the original retail purchaser, that the medium on which the APX program is recorded and any hardware accessories sold by APX are free from defects for thirty days from the date of purchase. Any applicable implied warranties, including warranties of merchantability and fitness for a particular purpose, are also limited to thirty days from the date of purchase. Some states don't allow limitations on a warranty's period, so this limitation might not apply to you. If you discover such a defect within the thirty-day period, call APX for a Return Authorization Number, and then return the product along with proof of purchase date to APX. We will repair or replace the product at our option.

You void this warranty if the APX product: (1) has been misused or shows signs of excessive wear; (2) has been damaged by use with non-ATARI products; or (3) has been serviced or modified by anyone other than an Authorized ATARI Service Center. Incidental and consequential damages are not covered by this warranty or by any implied warranty. Some states don't allow exclusion of incidental or consequential damages, so this exclusion might not apply to you.

DISCLAIMER OF WARRANTY AND LIABILITY ON COMPUTER PROGRAMS.

Most APX programs have been written by people not employed by Atari, Inc. The programs we select for APX offer something of value that we want to make available to ATARI Home Computer owners. To offer these programs to the widest number of people economically, we don't put APX products through rigorous testing. Therefore, APX produts are sold "as is", and we do not guarantee them in any way. In particular, we make no warranty, express or implied, including warranties of merchantability and fitness for a particular purpose. We are not liable for any losses or damages of any kind that result from use of an APX product.

ATARI PROGRAM EXCHANGE

REVIEW FORM

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 us 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. Thank you for helping us!

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 (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 user instructions and why?

____

____

11. Other comments about the software or user instructions:

____

____

____

____

____

____

STAMP

ATARI Program Exchange
Attn: Publications Dept.
P.O. Box 50047
60 E. Plumeria Drive
San Jose, CA 95150

[seal here]

BASIC CROSS-REFERENCE UTILITY
(XREF)

by

Lane Winner

CONTENTS

INTRODUCTION __ 2

Overview __ 2

Required accessories __ 2

Optional accessories __ 2

GETTING STARTED __ 3

USING XREF __ 4

Running the program __ 4

Output __ 4

Reading your source program __ 4

Generating the results __ 4

TROUBLESHOOTING __ 6

Error codes __ 6

Program operation warnings and limitations __ 6

INTRODUCTION

OVERVIEW

XREF is a utility program that gives you important information about your BASIC programs. It tells you the number of variables you've used, which is helpful in large programs, where you might be approaching the 128-variable limit. It also lists the names of all these variables and the lines in which you've used them. This information can be valuable for debugging complex BASIC programs, because it helps you to see where your variables are located and how they interact. Finally, it lists your constants and tells you how often you used each one.

The program is simple to use. You answer one prompt and XREF does all the work! You can then obtain a video display version or a printed version of the resulting information.

REQUIRED ACCESSORIES

40K RAM
ATARI BASIC Language Cartridge
ATARI 810 Disk Drive

OPTIONAL ACCESSORIES

ATARI printer or equivalent printer

GETTING STARTED

1. Insert the ATARI BASIC Language Cartridge in the (Left Cartridge) slot of your computer.

2. If you want printed output, turn on your printer. If you're using an ATARI 825 Printer, turn on your ATARI 850 Interface Module.

3. Turn on your disk drive(s) and insert the XREF diskette in disk drive one.

4. Power up your computer and turn on your video screen.

5. When the READY prompt displays, type one of the following commands: (1) to run the version that prints your output, type RUN "D:XREF.PRT" and press the RETURN key, or (2) to run the version that displays your output at your video screen, type RUN "D:XREF" and press the RETURN key.

Note. XREF requires that you store your file in SAVE format rather than in LIST format. If you've stored your file in LIST format, load it into RAM and then store it back to disk via the SAVE D:filename command.

USING XREF

RUNNING THE PROGRAM

After the COPYRIGHT 1981 ATARI notice appears, XREF prompts you for the file name of your BASIC program:

INPUT FILE NAME:

If you're using one disk drive, remove the XREF diskette and insert the diskette with the program you want cross-referenced, and then type in the device initial and the file name as it appears on the DOS directory (e.g., D:CHECKER). If you're using more than one disk drive, remember to specify the number of the drive containing your file (e.g., D2:CHECKER).

XREF then takes over. If you're using the program that prints your output, XREF will automatically print the results. If you're using the program that displays your output at the video screen, you'll need to stand by until XREF starts listing the results. Then use the CTRL/1 function to stop the display temporarily so that you can write it down; press CTRL/1 to resume the listing.

OUTPUT

Reading your source program

The first output will be to your video screen and will consist of what look like random characters. These are the Screen Editor's interpreation of code stored by the system in internal symbol tables. Next, XREF lists how many variables you used and the message TABLE SAVE IN PROGRESS indicates it is setting up the variable table. Then XREF slowly writes the source program on the screen. However, XREF changes the variables and constants as it sets them up for its own use. XREF also pauses now and then for a disk read. XREF obeys all control and escape commands it encounters, so don't worry if the cursor jumps around on the screen or if you hear a beep now and again. Once XREF has read the complete source file, it displays the message EOF FOUND. And when it has finished working with your source program, it displays the message END OF LIST.

Generating the results

Once XREF completes its analysis of your source program, it displays (or prints) the results. Figure 1 illustrates a sample output--video display or printed.

First it displays the CROSS-REFERENCE message, followed by the total number of variables used (1) and the variable table (2). Each variable name displays in the far left column; a left parenthesis after the name indicates a one- or two-dimensional array and a dollar sign after the name indicates a string; the line numbers in your souce program using the variable display following this information.

Notice that one variable, RAM, has no line numbers following it (3). Any variable you deleted from your program, but is still present in BASIC's internal symbol table displays in this form. To remove such variables from the internal symbol table, LIST your source program to disk and ENTER it back into RAM (this erases the current internal symbol table). Then SAVE your source program back on disk again (BASIC will then generate a new internal symbol table for your program). For more information about this procedure, read the sections on LOAD, SAVE, LIST, and ENTER in the ATARI Disk Operating System Reference Manual. The BASIC Reference Manual, pp. 2 - 3, also contains information about using the LIST format.

The XREF variable table is formatted as described above either on the display screen or on the printer, depending on which XREF program you're using. If you're using the video display version (file name XREF), the screen will be formatted as described. If you're using the printed version (file nam XREF.PRT), the screen display will be somewhat jumbled, but the printed table will be formatted as described.

Next, XREF generates the constant chart, which consists of two columns of numbers (4). The left column lists the number of times each constant is used; the right column lists the constant.

```
                    CROSS REFERENCE

(1)       4         VARIABLES USED

(2)       RAM(      10        30        60        70
                    80        90        140       140
                    140       140

          J         20        30        30        40
                    50        60        60        70
                    80        80        90        90
                    100       130       140       140
                    140       140       140       150
(3)       RAM

           X         60        60        70        70
                     80        80        90        90


(4)          NO.
            TIMES
            USED        CONSTANTS

             3             1
             2             2
             2             3
             2             4
             3             5
             3             6
             8             8
             1             48
```

Figure 1. Sample Output

TROUBLESHOOTING

ERROR CODES

You might encounter some error codes during start-up procedures. ERROR 170- (File Not Found) will display if you enter a file name in response to the INPUT FILE NAME: prompt that doesn't correspond to a file name in your DOS directory. Make sure you specified the correct disk drive and that you spelled the file name correctly. Then start over by typing RUN and answering file name the prompt.

ERROR 130- (Nonexistent Device specified. Device is not turned on or not attached.) will display if you forget to enter D: preceding your file name. Type RUN and re-enter your file name in response to the file name prompt.

ERROR 138- (Device Timeout. Device doesn't respond. Check connections between peripheral equipment and console.) will display is you're using the printer version of XREF but neglect to turn on your printer (and interface). Type RUN and start over. If you don't have a printer connected and you get this error code, then you're using the wrong program. Use the RUN "D:XREF" command and start again.

PROGRAM OPERATION WARNINGS AND LIMITATIONS

XREF takes a substantial amount of time to run, and you'll be frustrated if you've waited through a long program only to lose the result should someone accidentally press the BREAK key after the program has run or should you be called away from the computer. Or, then again, you might want to see the results another time. To reproduce the variable cross-reference and constant tables, type GOTO 3050, which will call the subroutine generating these tables.

LIMITED WARRANTY ON MEDIA AND HARDWARE ACCESSORIES.

We, Atari, Inc., guarantee to you, the original retail purchaser, that the medium on which the APX program is recorded and any hardware accessories sold by APX are free from defects for thirty days from the date of purchase. Any applicable implied warranties, including warranties of merchantability and fitness for a particular purpose, are also limited to thirty days from the date of purchase. Some states don't allow limitations on a warranty's period, so this limitation might not apply to you. If you discover such a defect within the thirty-day period, call APX for a Return Authorization Number, and then return the product along with proof of purchase date to APX. We will repair or replace the product at our option.

You void this warranty if the APX product: (1) has been misused or shows signs of excessive wear; (2) has been damaged by use with non-ATARI products; or (3) has been serviced or modified by anyone other than an Authorized ATARI Service Center. Incidental and consequential damages are not covered by this warranty or by any implied warranty. Some states don't allow exclusion of incidental or consequential damages, so this exclusion might not apply to you.

DISCLAIMER OF WARRANTY AND LIABILITY ON COMPUTER PROGRAMS.

Most APX programs have been written by people not employed by Atari, Inc. The programs we select for APX offer something of value that we want to make available to ATARI Home Computer owners. To offer these programs to the widest number of people economically, we don't put APX products through rigorous testing. Therefore, APX produts are sold "as is", and we do not guarantee them in any way. In particular, we make no warranty, express or implied, including warranties of merchantability and fitness for a particular purpose. We are not liable for any losses or damages of any kind that result from use of an APX product.

ATARI PROGRAM EXCHANGE

REVIEW FORM

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 us 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. Thank you for helping us!

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 (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 user instructions and why?

____

____

11. Other comments about the software or user instructions:

____

____

____

____

____

____

STAMP

ATARI Program Exchange
Attn: Publications Dept.
P.O. Box 50047
60 E. Plumeria Drive
San Jose, CA 95150

[seal here]

ULTIMATE RENUMBER UTILITY

by

Justin E. Wilder and Douglas J. Wilder

INTRODUCTION

OVERVIEW

The ULTIMATE RENUMBER UTILITY provides a convenient renumber function when you use ATARI BASIC with your Atari Home Computer System. It's easy and natural to use, because it operates like the built-in commands. You can change your BASIC program line numbers starting anywhere and renumbering to the end of the program. All you do is specify the starting line number in your BASIC program, the new number you want to assign this line, and your desired increment between line numbers. Or, you can accept the program's default values and simply type the direct mode command RENUMBER. Not only are numbers changed at the beginning of each line, but all references to these numbers are changed within any program statement to correspond to the new line numbers. The renumbering is carried out completely, but any reference line number not found among the lines to be changed is left as is. If renumbering cannot be completed because of insufficient memory, or if it would result in line numbers greater than ATARI BASIC allows, your program is left unchanged and the message "CANNOT RENUMBER" displays on your screen.

ULTIMATE RENUMBER UTILITY is entirely in machine language and it operates directly on a BASIC program in memory. Thus, it's very fast and it has no BASIC lines to appear with yours when you list or save your program. It reduces memory available for your BASIC program by only 599 bytes and is not affected by a NEW command or by a SYSTEM RESET.

REQUIRED EQUIPMENT

Cassette version
- 8K RAM
- ATARI 410 Program Recorder

Diskette version
- 16K RAM
- ATARI 810 Disk Drive

ATARI BASIC Language Cartridge (for your BASIC programs)

CONTACTING THE AUTHORS

Users wishing to contact the authors about ULTIMATE RENUMBER UTILITY may write to Justin E. Wilder at:

1322 E. Douglas Street
Goshen, IN 46526

GETTING STARTED

LOADING ULTIMATE RENUMBER UTILITY INTO COMPUTER MEMORY

1. If you have another program in memory, save it on cassette or diskette, because you must turn off your computer to load this utility.

2. If you have an ATARI 850 Interface Module, make sure it's turned off before you load in ULTIMATE RENUMBER UTILITY. You can turn it on later so that you can use a printer connected to the parallel port, but the RS232C ports won't be usable while this utility is in memory.

3. If you have the cassette version of ULTIMATE RENUMBER UTILITY:

   a. Have your computer turned OFF.

   b. Insert the ULTIMATE RENUMBER UTILITY cassette into the program recorder's cassette holder and press REWIND on the recorder until the tape rewinds completely. Then press PLAY to prepare the program recorder for loading the program.

   c. Turn on the computer while holding down the START key.

   d. When you hear a beep, release the START key and press the RETURN key. The program will load into computer memory and start automatically.

If you have the diskette version of ULTIMATE RENUMBER UTILITY:

   a. Have your computer turned OFF.

   b. Turn on your disk drive.

   c. When the BUSY light goes out, open the disk drive door and insert the ULTIMATE RENUMBER UTILITY diskette with the label in the lower right-hand corner nearest to you. (Use disk drive one if you have more than one drive.)

   d. Turn on your computer and your TV set. The program will load into computer memory.

THE DISPLAY SCREEN

When the program has loaded into computer memory, the READY prompt appears on your TV screen to tell you that the utility is in memory and ready to use whenever you need it.

USING A DIFFERENT FILE FOR THREE- OR FOUR-DISK SYSTEMS

Users having one or two disk drives use the AUTORUN.SYS file on the APX diskette, which is the utility's object file. (As is true for all APX diskettes, it's a good idea to duplicate this file onto another diskette and store the APX diskette as your backup.) The APX diskette also contains a file named RENUMBER.4DR. This file is the utility's object file for anyone using three or four disk drives. The current AUTORUN.SYS file sits in the wrong location in RAM for such expanded systems and needs to be relocated. If you have three or four drives, use DOS-II option C to copy the file to another diskette, renaming it AUTORUN.SYS as you copy it (the response to the option prompt is D:RENUMBER.4DR,D2:AUTORUN.SYS).

USING ULTIMATE RENUMBER UTILITY

RENUMBER COMMAND FORMAT

When you want to renumber a BASIC program in memory, type the RENUMBER command (you can abbreviate it to REN.--be sure to include the period). The command format is:

```
RENUMBER [lineno][,lineno [,incr]]
```

The first parameter is the line number at which you want the renumbering to start. The program renumbers all lines from this one to the end of your program. If your program doesn't have the line number you specify, the program starts renumbering at the next higher line number. The default value is zero.

The second parameter is the new line number you want to assign the first line to be renumbered. To avoid overlapping with lines that aren't renumbered, this parameter must be at least as great as the first parameter. If you do specify a value less than the first parameter, the program assigns it the same value as the first parameter. The default value is ten.

The third parameter is the increment you want used between line numbers of your renumbered program. If you specify zero, the program assigns this parameter a value of one. The default value is ten.

Specifying parameter values

All parameters are optional. The program uses the default value(s) for any parameter(s) you don't specify. For example, to renumber a program from line 120 to the end, using the default increment of 10, the command is REN.120 <RETURN>. By typing a comma instead of a value, you may omit parameters for which you want to use the default value and then specify the value(s) for the subsequent parameter(s). For example, to renumber a program from the beginning to the end, using 10 as the new starting line number and an increment of 5, the command is REN.,,5 <RETURN>. If you accept the utility's default values for all three parameters, then you need type only RENUMBER (or REN.) and press the RETURN key.

Numeric values for parameters

The program rounds any noninteger parameters you use to the nearest integer value. Any number greater than 32767 or less than 0 produces an error message.

CHECKING EXPRESSIONS AS LINE NUMBER REFERENCES

A powerful feature of ATARI BASIC is the use of expressions or variables in place of line number references. Since we can't be sure such references are correct after renumbering, any program lines with expression or variable line number references are listed on the screen after the renumbering process is completed. The utility displays the number of each line in which one is found, with the message "CHECK EXPRESSION IN LINE". The line number repeats for each statement containing such an expression in the line. Check all these expressions to make sure they're correct for the new line numbers. A feature of ULTIMATE RENUMBER UTILITY helps you to check: if the expression begins with a number, the program treats the first number in the expression as a tie to a specific line number and makes the corresponding change. For example, in the expression "RESTORE 300+10*A", the "300" is subject to renumbering but the "10" is not. Likewise, in "GOTO MARK+100" the "100" would not be changed because it isn't the first value in the expression. If you write programs with this feature in mind, your effort required to check such line references will be less, but you must still check any expression for a line number reference after using the renumber program.

If your program contains more than twenty statements with expressions for line numbers, the first twenty display on the screen, followed by the message "& MORE". To find the rest, you can temporarily insert "REM" between the line number and statement of some of these and renumber again, using the same parameters as before.

References to lines that don't exist remain unchanged.

MOVING BLOCKS OF CODE

You can move blocks of lines from one part of your program to another. To do so, LIST the part on cassette or diskette that you want to place earlier in the program. Next, renumber the lines above the new location to be higher than those being moved. Then ENTER the lines back into memory from the cassette or diskette. Finally, correct any line number references between the moved part and the part that was renumbered while it was on cassette or diskette and remove the extra lines by typing each line number and pressing the RETURN key.

IF PART OF RENUMBER PROGRAM IS LOST

If you lose part of the renumber utility because you called DOS without having a MEM.SAV file, or because you did a POKE in memory, the whole program disengages upon your next keyboard input. The screen clears and the program ignores your next line of BASIC from the keyboard, but your BASIC program is protected against a computer malfunction due to faulty machine code.

ADVANCED TECHNICAL INFORMATION

MEMORY AND VECTOR USAGE

The ULTIMATE RENUMBER UTILITY uses page six in RAM for subroutines. The boot operation starts loading in the last part of page five, putting all of the boot code where it will later be overwritten by BASIC so that it does not occupy memory when it is no longer of any use. The address in the MEMLO vector is changed, moving up the starting point for the BASIC program by 599 bytes, and the main renumber program is inserted in this space.

The list of line numbers to be changed starts just above the BASIC program in the area reserved for contents of dimensioned strings and arrays, and continues upward through the screen memory if needed. A GRAPHICS 0 command is executed as renumbering is competed to re-establish the display list and clear the screen of this data. The printer buffer is used to store numbers of the lines containing expressions for line number references. Other temporary data is stored in parts of RAM page zero used by BASIC in floating point calculations.

The keyboard interrupt vector is used to set up a return to the renumber search routine whenever a RETURN is processed by the operating system. When it is finished, the renumber routine returns control to BASIC with an RTS operation. The initialization vector is used to establish the MEMLO and keyboard interrupt vectors at startup and whenever SYSTEM RESET is pushed to make this program SYSTEM RESET-proof. The initialization routine near the end of page six is the only part executed after the renumber program is disengaged when some of the code is lost.

LIMITED WARRANTY ON MEDIA AND HARDWARE ACCESSORIES.

We, Atari, Inc., guarantee to you, the original retail purchaser, that the medium on which the APX program is recorded and any hardware accessories sold by APX are free from defects for thirty days from the date of purchase. Any applicable implied warranties, including warranties of merchantability and fitness for a particular purpose, are also limited to thirty days from the date of purchase. Some states don't allow limitations on a warranty's period, so this limitation might not apply to you. If you discover such a defect within the thirty-day period, call APX for a Return Authorization Number, and then return the product along with proof of purchase date to APX. We will repair or replace the product at our option.

You void this warranty if the APX product: (1) has been misused or shows signs of excessive wear; (2) has been damaged by use with non-ATARI products; or (3) has been serviced or modified by anyone other than an Authorized ATARI Service Center. Incidental and consequential damages are not covered by this warranty or by any implied warranty. Some states don't allow exclusion of incidental or consequential damages, so this exclusion might not apply to you.

DISCLAIMER OF WARRANTY AND LIABILITY ON COMPUTER PROGRAMS.

Most APX programs have been written by people not employed by Atari, Inc. The programs we select for APX offer something of value that we want to make available to ATARI Home Computer owners. To offer these programs to the widest number of people economically, we don't put APX products through rigorous testing. Therefore, APX produts are sold "as is", and we do not guarantee them in any way. In particular, we make no warranty, express or implied, including warranties of merchantability and fitness for a particular purpose. We are not liable for any losses or damages of any kind that result from use of an APX product.

ATARI PROGRAM EXCHANGE

REVIEW FORM

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 us 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. Thank you for helping us!

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 (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 user instructions and why?

____

____

11. Other comments about the software or user instructions:

____

____

____

____

____

____

|STAMP|

ATARI Program Exchange
Attn: Publications Dept.
P.O. Box 50047
60 E. Plumeria Drive
San Jose, CA 95150

[seal here]

VARIABLE CHANGER

by

Lane Winner

CONTENTS

INTRODUCTION __ 1

Overview __ 1

Required accessories __ 1

GETTING STARTED __ 2

USING THE PROGRAM __ 3

Entering input and output file names __ 3

Main Menu __ 3

1) NUMBER OF VARIABLES AND NAMES __ 4

2) CHANGE A VARIABLE NAME __ 4

3) CHANGE ALL VARIABLE NAMES __ 5

4) GENERATE TWO CHARACTER NAMES __ 5

5) GENERATE ONE CHARACTER NAMES __ 5

6) ALL NAMES THE SAME CHARACTER __ 6

7) CLOSE FILES AND EXIT __ 6

TROUBLESHOOTING __ 7

Program operation limitations and warnings __ 7

INTRODUCTION

OVERVIEW

[Scene 1.] Let's see...I'll name this variable A, and this variable B, and this one C, and this one....

[An hour later.] Hmmmm. Guess I'd better start back at A--I'll assign this variable AA and the next one BB, and the next one CC....

[Scene 2. Four months later]. I guess I'll make that little revision to my program while the kids are off my system! Let's see...oh yeah, variable A stood for my first counter, and B for....

[Two hours later.] Jeeeeez! These variables are driving me nuts! Why oh why didn't I assign them meaningful names? What I wouldn't give for a quick way to rename these gutless letters!

Relax. VARIABLE CHANGER is a painless way to repair the damage. In fact, this menu-driven program solves a number of potential problems involving variables. It can change a variable's name throughout your BASIC program by modifying your prgram's internal symbol table. You can either change selected variable names, or you can tell VARIABLE CHANGER to change all your variables to the shortest possible length. This latter option helps you when you need to compress a program into less space.

VARIABLE CHANGER can also help prevent your program from being pirated. It can convert all your variable names to the same non-printing character. Because the BASIC interpreter has already translated your variable names to an internal form, your program will still run, but no one will be able to understand it by using the LIST command, and so another user would have difficulty trying to modify your program.

REQUIRED ACCESSORIES

24K RAM
ATARI BASIC Language Cartridge
ATARI 810 Disk Drive

GETTING STARTED

1. Insert the ATARI BASIC Language Cartridge in the (Left Cartridge) slot of your computer.

2. Turn on your disk drive and insert the VARIABLE CHANGER diskette.

3. Power up your computer and turn on your video screen.

4. When the READY prompt displays, type RUN "D:VARCNR.APX" and press the RETURN key. The program will load into RAM and start.

NOTE. VARIABLE CHANGER requires that you store your file in SAVE format rather than in LIST format. If you've stored your file in LIST format, load it into RAM and then store it back to disk via the SAVE D:filename command.

USING THE PROGRAM

ENTERING INPUT AND OUTPUT FILE NAMES

After the COPYRIGHT 1981 ATARI notice displays, you see the prompt

```
ENTER INPUT FILENAME:
```

Enter the device code and file name of the program you want VARIABLE CHANGER to work on and press the RETURN key (e.g., D:ACCOUNT for a file on disk drive one, or D2:CHECKS for a file on disk drive two).

Then VARIABLE CHANGER asks you to indicate where you want to store your file after you've modified it. The prompt is:

```
ENTER OUTPUT FILENAME:
```

Enter the device code and file name in the same format you used for your input file. If you store your output file back on the same diskette, be sure to assign it a different name or you'll erase your original file. (To request a printout, type P: in response to the prompt; to display it on your video screen, type E: in response to the prompt; and to store it on cassette, type C: in response to the prompt.)

VARIABLE CHANGER then displays the Main Menu for you to select the function(s) you want to perform.

MAIN MENU

The Main Menu looks like this:

```
OPTIONS ARE:

    1) NUMBER OF VARIABLES AND NAMES

    2) CHANGE A VARIABLE NAME

    3) CHANGE ALL VARIABLE NAMES

    4) GENERATE TWO CHARACTER NAMES

    5) GENERATE ONE CHARACTER NAMES

    6) ALL NAMES THE SAME CHARACTER

    7) CLOSE FILES AND EXIT

SELECT OPTION:
```

Figure 1. Main Menu

Choose a menu selection by entering its corresponding number and pressing the RETURN key (e.g., type 2 + RETURN if you want to change a variable name).

1) NUMBER OF VARIABLES AND NAMES

Use this selection to display a list of all the variables in the internal symbol table and the number of variables in the table. The format is:

```
TD
ERR
A
OLDS
  :
31 VARIABLES FOUND

HIT RETURN
```

Remember that VARIABLE CHANGER looks at your program's internal symbol table and not at your program. Thus, the list might contain variables no longer active in your program but still in the table. To delete such variables, store the program in LIST format on diskette, type NEW (to erase the current symbol table), load your program back into RAM via the ENTER command, and then store it back on diskette in SAVE format (which causes the system to write a new table). For more information about this procedure, see the sections on LOAD, SAVE, LIST, and ENTER in the Disk Operating System Reference Manual.

Press the RETURN key to return to the Main Menu.

2) CHANGE A VARIABLE NAME

Use this selection to rename one variable. A prompt asks for the current name of the variable:

```
        OLD VARIABLE NAME:
```

Enter the current name (e.g., COUNTER1). A second prompt asks for the new name you want to assign this variable:

```
        NEW VARIABLE NAME:
```

Enter the new name (e.g., CTR1). VARIABLE CHANGER then renames this variable throughout your program. Press the RETURN key to return to the Main Menu.

3) CHANGE ALL VARIABLE NAMES

Use this selection to rename every variable in your program, one at a time. VARIABLE CHANGER displays each current name, and prompts you for the new name you wish to assign to this variable. If you press the RETURN key without enterng a new name, that name remains unchanged, and VARIABLE CHANGER prompts you for the next variable. For example to rename variables A and C, but leave variable BANK alone, the sequence would be as follows:

```
        OLD VARIABLE NAME: A
        NEW VARIABLE NAME: ACCOUNT

        OLD VARIABLE NAME: BANK
        NEW VARIABLE NAME: <RETURN>

        OLD VARIABLE NAME: C
        NEW VARIABLE NAME: CHARGE

            :

        HIT RETURN
```

Press the RETURN key to return to the Main Menu.

4) GENERATE TWO CHARACTER NAMES

Use this selection to tell VARIABLE CHANGER to assign two-character names to all your variables automatically. It assigns names starting with AA and proceeding sequentially to DX. This menu selection has no prompts. While VARIABLE CHANGER is reassigning names, the message COMPACTING displays on your video screen. When the HIT RETURN prompt displays, press the RETURN key to return to the Main Menu.

5) GENERATE ONE CHARACTER NAMES

Use this selection to tell VARIABLE CHANGER to assign one-character names to all your variables automatically. It assigns names in order of the ATASCII Character Set (see

Appendix C of the *BASIC Reference Manual* for a list of these characters). This menu selection has no prompts. While VARIABLE CHANGER is reassigning names, the message COMPACTING displays on your video screen. When the HIT RETURN prompt displays, press the RETURN key to return to the Main Menu.

6) ALL NAMES THE SAME CHARACTER

Use this selection to rename all your variables to the same character. Once you use this option, reading your code will be extremely difficult, so don't do it until your program is in final form. (Of course, should you need to revise your program later, you can always use the original file instead of the file you reworked with VARIABLE CHANGER.) A prompts asks you for the ATASCII number of the character you want as your universal name:

```
        ATASCII NUMBER OF CHARACTER:
```

Enter any ATASCII number (e.g., 32 for "blank" or 65 for "A") and press the RETURN key. While VARIABLE CHANGER is renaming your variables, the message COMPACTING displays on your video screen. When the HIT RETURN prompt displays, press the RETURN key to return to the Main Menu. From then on, all your variables will appear as that character when listed.

7) CLOSE FILES AND EXIT

Use this selection to end your session with VARIABLE CHANGER. The message PLEASE WAIT FOR READY MESSAGE displays while VARIABLE CHANGER stores your reworked program in SAVE format under the file name you specified when you began the session. Then the READY prompt displays, returning you to BASIC.

TROUBLESHOOTING

PROGRAM OPERATION LIMITATIONS AND WARNINGS

If you lengthen your variable names, you could cause a line to exceed the logical line limit of the Screen Editor. The program will still run, but you won't be able to edit these lines in the future because the Screen Editor will truncate them. Therefore, it's a good idea to look over lines containing any variables you intend to lengthen before doing so.

LIMITED WARRANTY ON MEDIA AND HARDWARE ACCESSORIES.

We, Atari, Inc., guarantee to you, the original retail purchaser, that the medium on which the APX program is recorded and any hardware accessories sold by APX are free from defects for thirty days from the date of purchase. Any applicable implied warranties, including warranties of merchantability and fitness for a particular purpose, are also limited to thirty days from the date of purchase. Some states don't allow limitations on a warranty's period, so this limitation might not apply to you. If you discover such a defect within the thirty-day period, call APX for a Return Authorization Number, and then return the product along with proof of purchase date to APX. We will repair or replace the product at our option.

You void this warranty if the APX product: (1) has been misused or shows signs of excessive wear; (2) has been damaged by use with non-ATARI products; or (3) has been serviced or modified by anyone other than an Authorized ATARI Service Center. Incidental and consequential damages are not covered by this warranty or by any implied warranty. Some states don't allow exclusion of incidental or consequential damages, so this exclusion might not apply to you.

DISCLAIMER OF WARRANTY AND LIABILITY ON COMPUTER PROGRAMS.

Most APX programs have been written by people not employed by Atari, Inc. The programs we select for APX offer something of value that we want to make available to ATARI Home Computer owners. To offer these programs to the widest number of people economically, we don't put APX products through rigorous testing. Therefore, APX produts are sold "as is", and we do not guarantee them in any way. In particular, we make no warranty, express or implied, including warranties of merchantability and fitness for a particular purpose. We are not liable for any losses or damages of any kind that result from use of an APX product.

ATARI PROGRAM EXCHANGE

REVIEW FORM

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 us 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. Thank you for helping us!

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 (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 user instructions and why?

11. Other comments about the software or user instructions:

STAMP

ATARI Program Exchange
Attn: Publications Dept.
P.O. Box 50047
60 E. Plumeria Drive
San Jose, CA 95150

[seal here]

LIMITED WARRANTY ON MEDIA AND HARDWARE ACCESSORIES.

We, Atari, Inc., guarantee to you, the original retail purchaser, that the medium on which the APX program is recorded and any hardware accessories sold by APX are free from defects for thirty days from the date of purchase. Any applicable implied warranties, including warranties of merchantability and fitness for a particular purpose, are also limited to thirty days from the date of purchase. Some states don't allow limitations on a warranty's period, so this limitation might not apply to you. If you discover such a defect within the thirty-day period, call APX for a Return Authorization Number, and then return the product along with proof of purchase date to APX. We will repair or replace the product at our option.

You void this warranty if the APX product: (1) has been misused or shows signs of excessive wear; (2) has been damaged by use with non-ATARI products; or (3) has been serviced or modified by anyone other than an Authorized ATARI Service Center. Incidental and consequential damages are not covered by this warranty or by any implied warranty. Some states don't allow exclusion of incidental or consequential damages, so this exclusion might not apply to you.

DISCLAIMER OF WARRANTY AND LIABILITY ON COMPUTER PROGRAMS.

Most APX programs have been written by people not employed by Atari, Inc. The programs we select for APX offer something of value that we want to make available to ATARI Home Computer owners. To offer these programs to the widest number of people economically, we don't put APX products through rigorous testing. Therefore, APX produts are sold "as is", and we do not guarantee them in any way. In particular, we make no warranty, express or implied, including warranties of merchantability and fitness for a particular purpose. We are not liable for any losses or damages of any kind that result from use of an APX product.

ATARI PROGRAM EXCHANGE

REVIEW FORM

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 us 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. Thank you for helping us!

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 (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 user instructions and why?

____

____

11. Other comments about the software or user instructions:

____

____

____

____

____

____

|STAMP|

ATARI Program Exchange
Attn: Publications Dept.
P.O. Box 50047
60 E. Plumeria Drive
San Jose, CA 95150

[seal here]