XBasher

Distributed by Genial Computerware

Contents

Genial Computerware
Box 183
Grafton, MA 01519

XBasher

Version 1.0

Introduction

XBasher is designed to reduce the size of an Extended BASIC program. It accomplishes this by shortening variable names, combining lines together, removing comments, and many other less-obvious techniques. XBasher has been designed with many options so that you can decide which of its many features you want to use on a particular program. XBasher may not seem fast; however, because of extensive use of assembly language, XBasher is between 50% and 200% faster then similar programs.

Preparing to use XBasher

XBasher works using the Merge file type. This means that before you can use XBasher you must first convert the program you wish to bash to MERGE format. To do this, load the program into Extended BASIC by typing:

```
OLD DSKn.filename
```

Once it is loaded into memory, you can save it out to disk in Merge format by typing:

```
SAVE DSKn.mergename,MERGE
```

When using XBasher you will be prompted for an Input file. At this prompt you should enter whatever you chose for "DSKn.mergename" above.

Running the Program

To run XBasher simply put the XBasher disk in drive one, and select Extended BASIC. It will start up automatically. There are actually two versions of XBasher on your disk. They are identical in function; however, one version runs with TI Extended BASIC while the other runs with MYARC Extended BASIC II. The LOAD program on the disk will automatically select the correct version. If you wish to run XBasher when you are already in TI Extended BASIC, type:

```
RUN "DSK1.XBASHER"
```

If you want to run XBasher when you are already in MYARC Extended BASIC II, type:

```
RUN "DSK1.XBASHERM"
```

The MYARC version loads the assembly language support off disk from the file DSK.XBASHER.XBASHERMO. The TI Extended BASIC file XBASHER has the assembly code merged in with the program so no separate file is required.

Using the Program

When you run XBasher you will first be presented with a title screen. Press any key to continue from this screen and you will be presented with the Option Screen. You can toggle any of these options on or off by simply hitting the letter at the beginning of the line. For example, hitting A will change "shorten variables" to "DON'T shorten variables." Hitting A again will change it back to "shorten variables." Each of the choices on this screen is explained below.

Shorten Variables: Selecting this option will cause XBasher to change all variable names in the program. It will change the first 26 numeric and the first 26 string variables to the letters A to Z and all variables beyond that to 2 letter combinations. This can result in a considerable memory savings, particularly if there are a lot of lengthy variable names in the program. Under the files menu, you can choose to have XBasher provide a list of all the variable names in the program and how they were changed.

Crunch Lines: This option will cause XBasher to combine lines together to save memory. XBasher will combine lines only where possible, so that program logic is not changed. Many times XBasher will create lines that are so long that they can not be entered normally at the keyboard; thus this option should be avoided for programs that are destined for publication in print.

Remove REMs and !s: This option tells XBasher to remove all comments from the program. If the comments are referenced with a GOTO, GOSUB, IF-THEN, or other such statement, XBasher will change the statement that references the comment so that the comment can be removed. The ability to remove comments, particularly those that are referenced by other statements, can save many bytes, especially in programs that were published in various magazines.

Remove LETs: The LET statement is not required by Extended BASIC, so this option allows you to have any LET statement removed from the program.

Change CALL CLEAR: This option changes all occurences of the CALL CLEAR statement to DISPLAY ERASE ALL. This actually saves 5 bytes for each CALL CLEAR replaced because of the way that Extended BASIC stores lines. Thanks to Aaron West for this useful trick.

DON'T Change SUB Digits: XBasher will change simple numeric constants in your program to variables to save space (more under Change Constants below). In general, however, you will not wish to invoke this option on Subprograms, as it will actually take up memory, rather then save memory. Unless the program has very long subprograms, you should leave this as "DON'T change SUB digits."

Change Constants: This option allows you to replace up to 5 single digit numeric constants with the variables @,\,[,], and _. Each replacement will result in a savings of 2 bytes because of the way Extended BASIC stores lines. XBasher will default to the following changes: @=0, \=1, [=2,]=3, and _=4. You can modify these by pressing a number from 1 to 5 on the option screen and entering the digit you wish. If you don't want XBasher to use a particular variable, replace the number with an "X."

When you have set up the option screen the way you want it, hit X to go to the files screen.

First you will be asked for an "Input file." At this prompt you should enter the name of the MERGE file that you created before running XBasher.

Next you will be prompted for "Output file." This is the name of the MERGE file that XBasher will create containing the bashed program. XBasher will automatically provide a default filename of the input filename you gave plus the letter "O" for output.

Finally you are asked for "Variable file." At this prompt hit ENTER (if you selected "shorten variables" on the option screen). If you want the listing, enter either a disk filename or the name of your printer, such as "PIO."

After you have entered the filenames, XBasher will start working. You will be presented with a Status Screen which contains various information about what XBasher is doing. XBasher makes two complete passes through your program. The first pass is to create variable lists, line number lists, and other such information. On the second pass XBasher actually makes the changes in your program. The status screen changes slightly from the first to second pass.

The first line of the status screen is "Line Count." This is simply the number of lines that XBasher has processed. The next line, "Processing Line", is the actual line number that XBasher is working on. "Last Variable Added" tells you the last variable that XBasher addded to its internal list of variables. In general this will change often at the beginning of a program and very little at the end. "Last Line Number Reference" tells what line number was referenced in the last GOTO, GOSUB, ON GOTO, IF-THEN, or other such statement. "Processing Segment" tells you whether XBasher is working on the "Main Program" or a subprogram. "Lines Printed to Disk" tells you how many lines of the bashed Extended BASIC program have been written out to disk. This is used only during the second pass. The "Last Variable Added" and "Last line number reference" are used only during the first pass.

When XBasher is finished, it will clear the screen and tell you to type:

```
NEW
MERGE DSKn.output
SAVE DSKn.filename
RUN
```

This will create a runnable file on disk of the bashed program. Before running XBasher on a program, you should make a back up of the program in case you want to go back to that version for some reason.

XBasher with XB/AL Programs

If you want to run XBasher on an Extended Basic program with assembly embedded in it, you will need to load it back differently. Save the file in MERGE format and run it through XBasher as you normally would. When XBasher is done, type:

```
NEW
```

Then load the original program into memory with:

```
OLD DSKn.original program name
```

Now load the DELLINES routine to delete all the lines:

```
MERGE DSK.XBASHER.DELLINE
RUN
```


Now load the output file and save it normally:

```
MERGE DSKn.output
SAVE DSKn.filename
RUN
```

The technique used in DELLINES to detect the start of the assembly code was designed by Tom Freeman of the LA 99ers.

XBasher with Pre-Scan It!

If you plan to run the program through both XBasher and Pre-Scan It!, it is better to run Pre-Scan It! first, as XBasher will compact any blank lines left by Pre-Scan It! If you do this, you should select the "Change Constants" option on either XBasher or Pre-Scan It!, but not both. Note also that as both Pre-Scan It! and XBasher will sometimes generate a line 1, you should type RES (resequence) after running it through Pre-Scan It!, before running it through XBasher.

Making a Back Up

If you make a back up copy of your XBasher disk, make sure to name the disk "XBASHER" because this is how XBasher finds its files on disk. You should make a back up copy of XBasher for your own protection, in case something should happen to the original disk. XBasher can be backed up using any disk manager program as it is not copy protected.

Assorted Notes

XBasher will add a line 1 to your program if you specify to "Change Constants." It may also add a line at the beginning of each subprogram if you specify "Change Subprogram Digits." Because of this it is best if you do a RES (resequence) before you create the MERGE file for XBasher to work on.

If there are user-defined subprograms in the program that you are running through XBasher, you must make sure that any one of the SUB statements is the first statement on a line, or XBasher will generate a fatal error.

Thank You

Thanks to Peter Hoddie for his numerous comments and suggestions on XBasher as it was developed, and to D.R. Fudge for testing XBasher as it was developed. Thanks also to Tom Freeman for the use of his technique for detecting A/L code embedded in a program, and to Todd Kaplan for ALSAVE, which enabled me to create the fast loader for the TI Extended Basic version.