

BASIC AID User's Manual

EIGEN SYSTEMS
P.O. BOX 10234
Austin, Texas 78766

INTRODUCTION

BASIC AID is a utility program designed to aid and assist Radio Shack's Color Basic and Extended Color Basic. The program, supplied in a ROM PACK, is plugged into the side of the Color Computer prior to turning on the power. The cartridge works with 4K, 16K, and 32K Color Computers. This manual describes BASIC AID's features and shows you how they can be used to make your programming life easier.

The BASIC AID User's Manual is divided into seven sections that describe how to use the program's commands. The seven sections are:

- o Installing BASIC AID
- o Using BASIC AID
- o Program Merging
- o Moving Program Segments
- o Automatic Line Numbering
- o Key Commands
- o Miscellaneous Functions

INSTALLING BASIC AID

BASIC AID consists of the ROM PAK cartridge, user's manual, and a plastic keyboard overlay. The overlay, which is placed over the keyboard, has BASIC AID's 7 commands and most of Extended Color Basic's commands printed on its surface. One command is printed by each key to show you the key that is pressed to generate that command. This procedure is covered in the section on Key Commands.

The cartridge plugs into the slot on the right side of the Color Computer. The cartridge should never be inserted when the computer's power is turned on. Otherwise, the components in the cartridge could be destroyed.

When BASIC AID is powered up, it reserves a block of memory at the top of the RAM memory map. This memory is used for internal variables, key definitions, etc. You must be careful not to issue a CLEAR command that conflicts with this memory area. If you alter the contents of this memory area, you'll destroy the contents of BASIC AID's internal data values. The program's reserved memory areas are listed below for the 32K, 16K, and 4K Color Computers:

32K system reserved from &H7F1C to end
16K system reserved from &H3F1C to end
4K system reserved from &HF1C to end

USING BASIC AID

With the cartridge plugged in and power applied, BASIC AID is automatically ready and waiting for your command! You don't need to enter an execute command or anything.

The commands and functions are accessed by simultaneously pressing a control key (the down arrow key) and the desired command key. In this manual, the down arrow key/command key sequence is indicated with the word 'Control' followed by a dash and the command key. As an example, the merge program command is shown as:

Control-1

In this example, the program would respond by beginning the program merging process (explained in the following section).

All BASIC AID commands are entered as shown above (substituting the actual down arrow key for the word 'Control') and used as described in the following sections. In addition, this manual shows the ENTER key as <ENTER>. In actual use, you would replace the <ENTER> by pressing the ENTER key.

PROGRAM MERGING

Would you like to have the convenience of disk utility programs or subroutine libraries but retain the low cost of cassette files? With the program merge command, you gain a lot of the convenience of disk without the additional cost!

You can load and merge Basic programs (or files) stored on tape with a Basic program currently in memory. During the merge, BASIC AID flashes the upper left corner of the screen so you'll know the file is being loaded.

Using the Command

You begin a merge operation by entering the command Control-1. The program prompts you for all input after the command is entered. The merge command's prompts and definitions are:

SLINE If you want to re-number the incoming program, enter the beginning line number following this prompt. If you don't specify a starting line number, BASIC AID assumes you want the incoming program's line numbers. Be careful with the line numbers! BASIC AID doesn't check for duplicate line numbers - it just overwrites the lines currently in memory!

, If you specified a beginning line number, the program prompts for the increment between line numbers.

FILE Specifies the file name to be merged. If no file name is entered, the program loads the first file found on the tape. The basic programs merged from tape must have been saved in standard format without the 'A' option. The merge command will not accept ASCII format files.

Error Conditions

During the merge operation, you can generate two types of errors. If the file being read in causes a read error, BASIC AID displays the message

'IO ERROR'

and terminates the operation. Secondly, if you attempt to load a program that exceeds your memory size, the program displays the message

'SN ERROR'

and terminates the operation. In both cases, BASIC AID doesn't merge anything with your program in memory.

Example

The following example illustrates a typical Merge operation. The example assumes a program or subroutine labeled BOX exists on tape. The angle brackets show material that you enter from the keyboard.

```
Control-1  
SLINE <100><ENTER>,<5><ENTER>  
FILE <BOX><ENTER>
```

After the final carriage return, the program searches for and loads the file you specified. In this example, the file is merged with your current program beginning with line 100. All branch instructions within the merged file being loaded are changed to reflect the new numbering sequence.

MOVING PROGRAM SEGMENTS

This command solves the age-old problem of restructuring a Basic program so it looks and performs better. If you've ever relocated 40 or 50 lines of code by re-typing them, you'll like the move command.

The command relocates a block of code from one part of the program to another. In the process of moving the code, you can re-number the lines and change the increment between the lines. During the move, BASIC AID searches for and updates all references to line numbers within that section of code.

You should avoid setting up duplicate line numbers during the move. If you specify a new series of line numbers that already exist, BASIC AID destroys the existing lines of code during the move. Always check your line numbers before starting the move operation.

Using the Command

The move command is specified by entering a Control-2. The program prompts you for all input after you enter the command. The merge command's prompts and definitions are:

- FROM The command wants the beginning line number of the section of code to be moved.
- , The command wants the last line number of the section to be moved.
- TO Enter the new starting line number. This is the location where the section of code will be relocated. Remember to check for duplicate line numbers before continuing!
- INC - Enter the desired increment between the line numbers for the new section of code.

Example

The following example illustrates a typical move operation. The example assumes a program exists in memory. The angle brackets show material that you enter from the keyboard.

```
Control-2 <ENTER>  
FROM <100><ENTER> , <200><ENTER>  
TO <30><ENTER> INC <5><ENTER>
```

After the final carriage return, the program moves the specified block of code from its current location to the new location. In the process of moving the code, the program changes all references to the old line numbers to reflect the new line numbers. BASIC AID not only changes branches within the section of code, but changes all references throughout the entire program to reflect the new line numbers.

AUTOMATIC LINE NUMBERING

BASIC AID resolves a big deficiency of Color Basic - automatic line numbering! It's amazing how much time you can save by not entering each line number.

The program contains two line numbering commands: Control-6 is used to enable or disable the automatic line numbering, while Control-7 is used to pre-enter the starting line number and increment.

Setting Up Line Numbering

After entering a Control-7, BASIC AID prompts you for the beginning line number and increment. The command's prompts and definitions are:

SLINE The command expects you to enter the beginning line number. If you press the ENTER key without entering a line number, the program aborts the operation.

, When the comma appears, you can enter the desired increment between each line number. If you don't enter a new increment, the program re-uses any previous increment or the default of 10.

Example

The following example illustrates the procedure for setting the beginning line number and desired increment. The angle brackets show material that you enter from the keyboard.

```
Control-7  
SLINE <25><ENTER> , <10><ENTER>
```

After the final carriage return, BASIC AID initializes the line count to the number 25 and the increment to 10. The automatic line numbering doesn't begin until you enter a Control-6 (described below).

Enabling/Disabling Line Numbering

When you're ready to start or stop automatic line numbering, enter a Control-6 from the keyboard. The first time you enter the command, BASIC AID enables the automatic line numbering feature. Each time you enter the command after that, it toggles the automatic line numbering on and off. When you're entering a Basic command line and want to turn off the automatic line numbering, enter the Control-6 before pressing ENTER. This method provides a clean break in the numbering sequence and permits you to re-enable line numbering later at the line following the the last one.

KEY COMMANDS

Normally, Basic programs require many commands and keywords to be entered from the keyboard - a very tedious process. BASIC AID provides two-key entry of most Extended Color Basic commands. The program assigns Basic commands to the keyboard as shown on the keyboard overlay.

BASIC AID permits you to enter two-key commands, re-define any Basic command key, restore a command key's original definition, and to save your key definitions on cassette so they can be used another time.

Entering Two-Key Commands

When you need a command or expression that is assigned to one of the keys, enter the control key followed by the desired command key. If you'll refer to the overlay, the Basic command PEEK is entered by pressing a Control-P. BASIC AID then automatically enters the command for you. The program will automatically enter anything that you have assigned to the keys.

Re-defining Keys

You can redefine one key each time you enter a Control-5. The prompts and definitions issued by BASIC AID are:

- KEY The program expects you to enter the key that will be redefined without pressing the control key. You can enter any key except keys 1 through 7. These keys define the programs major commands and can't be re-defined.
- CMD The program now expects you to enter the desired command or expression that will be assigned to the key. The new definition is terminated with the BREAK key.

Example

If you were to enter the following example, BASIC AID would assign the specified expression to the key. The characters between the angle brackets shows the material you would enter.

```
Control-5  
KEY <CLEAR> CMD <PRINT "THE VALUE IS";AB <ENTER>> <BREAK>
```

After you assigned this expression to the CLEAR key, you can enter a Control-CLEAR and BASIC AID automatically prints the expression and displays the value of the variable AB.

Retrieving Original Key Commands

You can retrieve the command originally assigned to a key at any time. After entering the Control-5, press the desired key in response to BASIC AID's prompt KEY. When the program requests the new command, press the BREAK key. The program will then restore the original command to the key.

Saving User-Defined Key Definitions

When you're ready to end a work session and have re-defined one or more command keys, BASIC AID permits you to save your definitions on cassette. Thus, you can define specialized command keys and re-use them repeatedly. The definitions are saved with the CSAVEM command. The address ranges and command formats for the three Color Computer models are shown below. In each case, the 'file' is replaced with your specific file name.

```
32K System: CSAVEM 'file', &H7F1C, &H7F98, &H7F1C
16K System: CSAVEM 'file', &H3F1C, &H3F98, &H3F1C
4K System:  CSAVEM 'file', &HF1C, &HF98, &HF1C
```

Enlarge/Compress User Command Area

BASIC AID initially reserves a 127 byte table for your command key definitions. Within this table, you're allowed to re-define as many command keys as desired. If this table size is too large or small, you can change it as described below.

The table contains the contents of BASIC AID's internal variables, a pointer to the beginning of your command definition table, and the actual table itself. You can alter the size of the table by changing the address contained in the pointer. The following addresses are for a 32K Color Computer:

```
&H7F99-&H7FFF BASIC AID's variables/etc.
&H7F97-&H7F98 PTR (pointer to user table)
&H7F1C       SOY (start of user table)
```

For a 16K system, PTR's memory location is &H3F97. For a 4K system, PTR's memory location is &HF97. The procedure to change the table size is:

1. Protect the area of memory you want to use with Basic's CLEAR command. If your new table will begin at address &H7000, you need to enter CLEAR n, &H6FFF (n is the string space your program needs).

For the following steps, enter the commands without using BASIC AID.

2. Using the POKE command, store a value of 0 (zero) in the first location of your new command table.

3. With the POKE command again, store the beginning address of your new table in the location PTR (described above).

Your table size has now been changed so it begins at the address you stored at PTR.

MISCELLANEOUS FUNCTIONS

BASIC AID contains two additional commands: the on/off and blank suppress commands. You can selectively toggle BASIC AID on or off by entering a Control-3. When you initially use BASIC AID, it is enabled.

When you enter two-key Basic commands and keywords, BASIC AID automatically inserts spaces after the commands to improve program readability. You can enable or disable this feature by entering a Control-4. The default condition of this command is disabled.