

## BUSINESS BASIC Addendum

### INTRODUCTION

This documentation is meant to serve as an addendum to the documentation for Business Basic ver. 1.0; its function is merely to describe the changes that were made in ver. 1.0 to create ver 2.0.

### SCREEN ROUTINES

The screen routines have been rewritten because of a restriction imposed by the original routines: a character could not be displayed in the lower right corner of the screen without a scroll occurring. This effectively resulted in the 1024 character display being reduced to a 1023 character display. The problem has been corrected by not scrolling until the 1025th character is written. This should have no effect on any previously written BASIC program.

In rewriting the routines, one additional feature was added: pressing the "control A" key when characters are being displayed will cause the system to pause until the key is no longer being pressed. This is useful for "freezing" the display while listing a program.

### LENGTH ERROR:

The "LENGTH ERROR" message will no longer be displayed. Instead, when the maximum number of characters have been entered, no more characters (except RUB, carriage return, ESC, or control X) will be accepted.

### LINE

The LINE command will now be accepted as an executable statement of the form: LINE expr. Example:

```
1390 IF A>L THEN LINE 132 ELSE LINE A
```

### DELAY

The DELAY command now has the format: DELAY [expr]. (Previously, the format was DELAY [constant]).

### DEL COMMAND:

The DEL command issued to delete a range of lines from the resident program. The format is: DEL <starting line>,<ending line>. Both starting and ending line numbers must be specified and must exist in the program. Example:

```
DEL 20,60
```

This would delete lines 20 through 60. Note: Using the DEL command causes the variable table to be cleared.

## BUSINESS BASIC Addendum

### RUN, SAVE, and LOAD:

In the RUN, SAVE, and LOAD commands, the drive number is now optional; if no drive number is specified, drive zero (0) will be accessed. Example:

```
RUN "TEST"
```

This is equivalent to RUN#0, "TEST".

### COMMON VARIABLES BETWEEN PROGRAMS:

This option allows one program to RUN another without destroying any previously created variables. Any user defined functions in the first program will be cleared; functions common to both programs will have to be defined in both programs. If the printer is OPEN when the RUN occurs, it will remain OPEN. The format of this command is:

```
RUN![#expr,]str expr
```

Where [#expr,] is the optional drive number and "str expr" is the name of the program to be RUN.

### FORMATTED INPUT:

Formatted input is achieved by using the ENTER KEYIN statement. It is essentially a method of "screening" keyboard input as it is being typed. Unwanted characters are simply ignored. This may be used to input such things as dates, ID numbers, or dollar amounts. The format is:

```
10 ENTER KEYIN expr1,expr2;str variable
```

Expr1 is the field type. This specified what type of input is to be expected, and must be an integer from 0 to 6. Field types are:

- 0 = any ASCII character string
- 1 = Date (format MM/DD/YY; slashed are automatic)
- 2 = Dollar amount (may be negative)
- 3 = Digits (accepts only 0-9)
- 4 = Any ASCII character except carriage return
- 5 = Digits (accepts 0-9 or carriage return)
- 6 = Dollar amount (must be positive)

Expr2 is the length of the field, where  $1 \leq \text{expr2} \leq \text{LINE LENGTH}$ . If  $\text{expr2} > \text{LINE LENGTH}$ , expr2 defaults to LINE LENGTH.

Str variable will contain the returned field. If str variable is a substring (i.e., A\$(2,6)), it will be buffered with spaces if the input field is shorter than the substring.

If any control character (except carriage return) is typed as the first character of a field, that character will be stored as a string variable, and control will be passed on to the next statement. If carriage return is typed as the first character of the field, str variable will be returned null (length zero) and control will be passed on to the next statement.

## BUSINESS BASIC Addendum

### Examples:

```
10 ENTER KEYIN 1,8;A$
20 ENTER KEYIN 3,5;N$
30 ENTER KEYIN 4,2;S$
```

Line 10 would be used to enter a date into A\$. Line 20 would be used to enter a five digit ID number into N\$. Note that since carriage return is ignored by field type 3, the returned value could contain no less than the specified (in this case five) number of digits. Line 30 is similar, except that input is alphanumeric; in this case, S\$ might contain a two letter state abbreviation ( such as CA).

### RESTORE ERROR

The RESTORE ERROR statement is used to restore ERROR control to BASIC after the execution of an ON ERROR statement.

### Example:

```
10 ON ERROR (E$,N$) GOTO 9000
20 PRINT "Enter code number:";
30 ENTER 1000,X$,X
40 PRINT
50 C=VAL(X$)
60 RESTORE ERROR
70 PRINT LOG(-1)
80 STOP
9000 PRINT "Invalid code number."
9010 GOTO 20
9999 END
RUN
```

```
Enter code number : XYZ
Invalid code number.
Enter code number: 123
OUT OF BOUNDS VALUE ERROR IN LINE 70
```

### DIRECTORY FUNCTION:

The DIR(expr) function is a string function which returns as its value, the directory of the specified drive. Expr must be an integer from 0 to 7.

### EXAMPLE

```
10 DIM A$(1024)
20 A$=DIR(0)
30 PRINT DIR(0) + CHR$(14); "Number of entries =";LEN(A$)/16
40 END
RUN
TEST .BA 1 BASIC .GO 79
Number of entries = 2
```

## LINE EDITOR

The line editor is used to change parts of a statement without having to retype the entire line. The editor is invoked by typing a dash followed by the number of the desired line. For example, to edit line 10 type:

-10

If there were a line 10, the line would be listed on the screen with a flashing cursor at the beginning of the line; however, if line 10 did not exist, no action would be taken. Once in the edit mode, the typing of the following keys would result in the given action:

control I - Move cursor right

control L - Move cursor left

control J - Move cursor to end of line

control X - Exit edit mode and cancel line(does not alter statement).

RUB - Delete one character

return - Exit edit mode. Input is taken as all characters in the line to the left of the cursor.

ESC - Exit to the resident operating system.

Any other character would simply be inserted into the line at the point indicated by the cursor. Note: Holding down any key for more than half a second will result in that character repeating at a rate of about ten characters per second.

The editor has been set up to operate with Digital Group keyboards, which do not have a latched output. In order to make the editor work with latched output keyboards, it may be necessary to make the following changes to BASIC:

46A6: CB

46B7: CB

46DD: CB

4725: CB

This may, in some cases(depending on strobe length), disable the automatic repeat function.