
TABULAR SYNOPSIS

OFFICIAL NOTICES	2
THE POWER OF Intel-BEST 3.3	3
HOW Intel-BEST 3.3 WORKS	3
HOW TO LOAD Intel-BEST 3.3	3
WARNINGS AND PRECAUTIONS	5
THE COMMAND SHORTCUTS	5
THE NEW PROMPT COMMANDS	6
THE MODIFIED NEW COMMAND	6
HOW TO MERGE BASIC PROGRAMS	7
A MISCELLANY OF FEATURES	7
THE NEW RELATIVE RESTORE COMMAND	9
THE NEW Ln8 COMMAND	10
THE NEW PR#2 COMMAND	10
THE COLOR TABLES ARE CORRECTED	10
HOW TO CHANGE BACKGROUND COLORS	11
HOW TO CHANGE NORMAL AND INVERSE COLORS	11
INSTANT SCREEN COLOR ROUTINES	13
MUSIC FUNDAMENTALS	13
ADAM'S MUSIC FUNDAMENTALS	14
THE NEW VOICE COMMANDS	15
THE NEW NOISE COMMANDS	17
SOUND COMMAND PROGRAMS	17
TEXT BLOCK GRAPHICS	21
THE BLOCK READ/WRITE ROUTINES	21
ERROR MESSAGES	22
HOW TO COPY Intel-BEST TO A BASIC BACKUP	23
WHAT ABOUT COMPATIBILITY	23

The software storage medium (datapak or disk) on which Intel-BEST 3.3 is stored is protected from the BASIC operating system (OS) command INIT. Each individual program is protected from DELETE and UNLOCK. We have incorporated these precautions for your protection.

OFFICIAL NOTICES**PRODUCT COPYRIGHT NOTICE:**

The software enclosed herein and this accompanying documentation are copyrighted by DIGITAL EXPRESS, INC. (DEI) 1986. All rights are reserved. Except as noted below, the software and/or the documentation may not, in whole or in part, be stored in any retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopy, recording, or otherwise without the prior written permission of DEI, the publisher and copyright owner.

SUBSIDIARY COPYRIGHTS AND TRADEMARKS:

SmartBASIC copyright 1983 by Lazer Microsystems, Inc. All rights reserved. ColecoVision is a registered trademark and ADAM, SmartBASIC, and SmartWriter are trademarks of Coleco Industries, Inc.

YOUR LEGAL RIGHT TO A BACKUP COPY:

DEI licenses to you, as the original purchaser, the right to make a back-up copy of the accompanying software for your personal use. Be advised that it is illegal for you to distribute any 'copies' to unlicensed users by sale, trade, or gift. By U.S. Copyright Law (TITLE 17, UNITED STATES CODE), unauthorized reproduction and/or sales may result in imprisonment or up to one year and fines of up to \$10,000 (17 USC 506). Copyright infringers may also be subject to civil liabilities.

DISCLAIMER NOTICE:

DIGITAL EXPRESS, INC. and/or the author of the enclosed software and this accompanying documentation have exercised due care in the preparation of this documentaion and the software which it describes. No warranty, either express or implied, is given as to the accuracy or suitability of the software or the documentation for a particular purpose and neither party shall assume any liabilities for consequential damages arising as a result of using the software or interpreting the documentation.

THE CORRECT REVISION OF SmartBASIC:

The enclosed software is designed specifically to for use with revision '79' SmartBASIC V1.0. To determine your revision: from within SmartBASIC enter PRINT PEEK(260). If you do not have the correct revision number, please contact Coleco Industries, Inc.

FULL WARRANTY ON THE STORAGE MEDIUM:

DEI warrants to the purchaser of this product that the storage medium is free from defective materials and workmanship. If the original storage medium (datapak or disk) fails to function properly, it will be repaired or replaced if returned to the address below:

DIGITAL EXPRESS, INC.

Route one, Box 29-6
Oak Hill, WV 25901

THE POWER OF Intel-BEST 3.3

Have you ever wondered how to use music in your own programs? How many times have you been annoyed by the DATA and REM space-bumps? Have you ever wanted to RESTORE to a specific line number? Are you tired of having to use three different color code tables? Would you like to be able to use more command shortcuts (similar to the question mark for the PRINT command)? Would like to have eight TEXT lines in the GR and HGR modes? Intel-BEST 3.3 will instantly implement all of these coveted features . . . and many more too!

Intel-BEST 3.3 accomplishes all of this by turning SmartBASIC V1.0 into a much more powerful language. This software documentation not only explains all of the program's features and how to use them; but, it also reveals some of SmartBASIC's powerful POKE addresses. In hardly no time at all, you'll be able to write very impressive BASIC programs.

HOW Intel-BEST 3.3 WORKS

Intel-BEST 3.3 is a pure Z-80 machine code program. It is 1742 bytes in length. In less than one second it changes more than 1000 standard SmartBASIC V1.0 addresses.

SmartBASIC V1.0 sets LOMEM to 27407. Intel-BEST only uses 193 bytes beyond this standard setting, ie, LOMEM is set to 27600. Thus, it hardly infringes upon BASIC's user RAM.

Intel-BEST rearranges and/or relocates several of BASIC's control tables. It also changes the default value of many of BASIC's standard addresses. To conserve memory, Intel-BEST uses nearly all of the gaps of non-functioning code within the interpreter's area of RAM.

HOW TO LOAD Intel-BEST 3.3

Getting this powerful machine code enhancement to SmartBASIC V1.0 into memory is very simple. We have provided two LOADING techniques for your convenience. Just use the procedure that best suits your needs. Please note that if you have more than one drive, you may use the appropriate drive number suffix.

LOADING PROCEDURE #1:

1. LOAD SmartBASIC V1.0 according to Coleco's instructions.
2. Remove the SmartBASIC datapak when it stops spinning.
3. Insert the Intel-BEST 3.3 datapak into the drive.
4. Type this:
RUN HELLO (then press [RETURN])
(be sure to type the letters correctly)
5. When the tape stops spinning, remove the datapak.

LOADING PROCEDURE #2:

1. LOAD SmartBASIC V1.0 according to Coleco's instructions.
2. Remove the SmartBASIC datapak when it stops spinning.
3. Insert the Intel-BEST 3.3 datapak into the drive.
4. Type this:
LOMEM:30000 (then press [RETURN])
(be sure to type each character correctly)
5. Type this:
BRUN Intel-BEST,A27600 (then press [RETURN])
(be sure to type each character correctly)
6. When the tape stops spinning, remove the datapak.
7. Type this:
LOMEM:27600 (then press [RETURN])
(be sure to type each character correctly)

The first procedure listed above uses the HELLO program to BRUN Intel-BEST 3.3. After Intel-BEST is active, the program will then attempt to RUN a program named FIRST. This is so that you can create an integrated software package, ie, one program executing another. You will probably want to use a file selection program as FIRST. Please LIST and study HELLO. You can easily modify the program to suit your specific needs. Please note that the HELLO program can not include any of the Intel-BEST 3.3 features. Only programs that you LOAD/RUN or enter after BRUNing Intel-BEST can incorporate Intel-BEST enhancements.

WARNINGS AND PRECAUTIONS

- All of the tape handling admonitions and precautions outlined in the user's guide that came with your ADAM apply to this storage medium.
 - The software storage medium (datapak or disk) on which Intel-BEST 3.3 is stored is protected from the BASIC operating system (OS) command INIT. The program itself is protected from DELETE and UNLOCK. We have incorporated these precautions for your protection. However, a media editor written in machine code may be used to override these failsafes.
 - LOMEM can not be set lower than 27600. If you attempt to do so, you'll get an error message.
 - Enter POKE and CALL addresses VERY carefully. If you enter an incorrect address, ADAM could CRASH. In this event, you'll have to reload SmartBASIC.
 - Always SAVE inchoate (partially completed) programs frequently. This will protect you from unexpected power outages, etc.
 - Programs that use the Intel-BEST 3.3 features will not execute properly unless you have Intel-BEST active in memory.
 - Intel-BEST 3.3 is upwardly compatible with SmartBASIC V1.0. Just about all SmartBASIC programs can be RUN with Intel-BEST. However, standard SmartBASIC V1.0 will not execute most Intel-BEST 3.3 programs.
 - Getting the maximum benefit from Intel-BEST 3.3 requires a working knowledge of SmartBASIC. This documentation is not intended to supplement your understanding of BASIC programming. Rather, its purpose is to help you use Intel-BEST to its fullest potential.
-

THE COMMAND SHORTCUTS

Nearly all versions of BASIC (including SmartBASIC) permit the question mark as a shortcut for the PRINT command. When the program is LISTed, all the question marks are converted to PRINT. Intel-BEST 3.3 includes five more similar shortcuts. This makes entering BASIC programs a lot easier and faster.

If you try to use one of the shortcut commands as a variable name, you'll get an error message just as if you had typed all the letters of the actual command. As with the '?' shortcut for PRINT, each of these new shortcuts will be converted to full spelling when you LIST the program.

The new shortcut commands are:

- F for FLASH
- H for HOME
- I for INVERSE
- N for NORMAL
- T for TEXT

THE NEW PROMPT COMMANDS

With standard SmartBASIC V1.0 the prompt is a right bracket '['. As soon as Intel-BEST is implemented, the prompt is changed to a greater than symbol '>'. We have selected this feature so that you can easily differentiate between SmartBASIC and Intel-BEST.

Intel-BEST also includes two new commands that affect the prompt. 'BEL' causes a bell to sound each time the prompt is displayed. 'NB' turns off the bell feature. When Intel-BEST is first executed, the prompt is in the 'BEL' mode. If you work with the music commands in immediate programming mode (without line numbers), you may need to turn the bell off because the bell automatically turns off the first voice.

Obviously, these two new commands are intended to benefit you as a programmer and not the users of your programs. If you choose to use either of these prompt commands in a program, you'll need to precede them with a PRINT CHR\$(4) statement.

In a program you might (for some very specialized purpose) want to turn on the bell. To do so, consider the following example. To turn the bell off, just replace 'BEL' with 'NB'.

```
10 PRINT CHR$(4); " BEL"
```

When you use these commands in immediate programming mode (as they are intended), just enter the command and press [RETURN]. For example:

```
BEL (then press [RETURN]) or,  
NB (then press [RETURN])
```

THE MODIFIED NEW COMMAND

To further assist you in differentiating between SmartBASIC and Intel-BEST, we have modified the NEW command. When you issue this NEW command, Intel-BEST will issue a TEXT command, display a title header, and then clear user RAM for another program. The message is:

```
Intel-BEST 3.3 by DEI (c) 1986
```

HOW TO MERGE BASIC PROGRAMS

How would you like to be able to merge BASIC programs directly from BASIC? SmartWriter provides a facility for this, but it takes a lot of unnecessary time switching between SmartWriter and SmartBASIC.

To merge BASIC programs from BASIC all you need to do is disable the part of the NEW command that clears user RAM. The machine code routine that controls this function starts at address 6356. This address normally contains a 205. If you POKE a 201 into 6356, you'll effectively disable NEW. With NEW inoperative you can now use the LOAD 'filename' command as if it were MERGE 'filename'.

You may find this trick very useful when you want combine various subprograms and routines. However, you should note that if two line numbers are the same, the last one LOADED will replace the previous one. You may want to renumber your routines before using this technique.

1. POKE 6356,201 (then press [RETURN])
 2. LOAD 'filename' (then press [RETURN])
 3. LOAD 'filename' (then press [RETURN])
 4. Continue LOADING files until you are done.
Note that if two line numbers are the same,
the last one LOADED will replace the previous one.
 5. POKE 6356,205 (then press [RETURN])
-

A MISCELLANY OF FEATURES

POKE limit reset:

SmartBASIC V1.0 will not permit POKES above address 54160. Intel-BEST 3.3 resets the POKE limit to 65535 (the highest standard memory address). You should not use the POKE command unless you have a thorough understanding of what the results will be. Intel-BEST uses addresses 65320 through 65516, inclusive.

How to switch to SmartWriter:

You can go directly to SmartWriter from BASIC with a simple CALL. Unlike pulling the reset switch, this technique will not reset any storage devices. When you implement this feature, the program in user RAM will be erased. USE IT VERY CAREFULLY!

To switch to SmartWriter, use either of the following CALLs:

CALL 64743 or,
CALL 64148

How to disable CNTL-C:

The ASCII value of CNTL-C is typically '3'. Address 16134 contains the ASCII value that enables the CNTL-C function. You can change it to any value that you want. If you POKE a 255 into 16134, you'll effectively disable CNTL-C because there is no keypress that corresponds to the ASCII value 255.

Increased program line length:

Standard SmartBASIC limits the length a program line that you enter to 128 characters, ie, four screen lines plus four characters. Intel-BEST sets the limit to 216 characters, ie, seven screen lines. With this feature, you can enter several more commands on each line and thus save programmable memory. Because BASIC adds spaces when you list a program, you'll probably want to limit your program lines to six screen lines.

The comma PRINT fields are modified:

Standard BASIC allows you to use the comma within a PRINT command's parameter to segregate the screen into two fields. Intel-BEST changes this comma feature. With Intel-BEST, the comma separates the screen into four fields.

Playing with the cursor:

Address 16953 contains the ASCII value of the cursor. It is typically '95', ie, an underscore. You can turn the cursor off by POKEing a '0' or a '32' into 16953. Address 17291 contains the value that controls the cursor's blink speed. It is typically set to '4'. Values '1' through '3' will speed up the blinking. Any other value will slow it down.

Unnecessary LIST spaces eliminated:

Intel-BEST 3.3 eliminates the extra space added after a semi-colon and a comma. This feature is included so that a LIST will more closely resemble the program as you typed it in. If you POKE a '0' into address 16148 and then LIST your BASIC program, you'll see it exactly as SmartWriter would display it. This address typically contains the ASCII value '32', ie, a blank space.

The DATA and REM space-bump bug is fixed:

With standard SmartBASIC V1.0, any time you LOAD a program or edit a line that contains a DATA or REM statement, BASIC adds an extra space after the command. If you have much data on a program line, it can be pushed off into oblivion -- never to be recovered. Intel-BEST corrects this bug!!! Finally, SmartBASIC V1.0's most irritating bug is fixed!!!

THE NEW RELATIVE RESTORE COMMAND

Data stored within a program can be assigned to variables by dint of the READ command. This command simply begins READING DATA in sequential fashion using a pointer to accomplish the task. READING starts at the first line number and continues until a DATA statement is encountered. Thereafter, the DATA pointer begins with the line number where the last READ was performed. If there are no more DATA statements, an error message will result. To circumvent this, a BASIC programmer can use the RESTORE command. RESTORE causes this DATA pointer to return to the first program line number and begin READING all over again. Thus, with RESTORE, you can use data stored in DATA statements more than once.

If you want to access a particular set of DATA, you have to use dummy variables to skip over unwanted data values. This cumbersome process inevitably delays program execution.

Because many advanced BASIC programmers are annoyed by this shortcoming, some sophisticated versions of BASIC include a command to control the DATA pointer. Unfortunately, SmartBASIC does not include such a feature. Intel-BEST 3.3 to the rescue!!!

Intel-BEST creates the new command 'LINE'. With this LINE command, you can set the DATA pointer to whichever line number that you specify. You use this command in the same manner as you use GOTO and GOSUB.

The LINE command vastly increases the speed of READING DATA because BASIC no longer has to needlessly search every program line to see if it contains a DATA statement. For optimum results you should use the LINE command just prior to a READ statement. Please note that, if you wish, you can still use the old RESTORE command.

The following simple program demonstrates how to use the new LINE command.

```
10 REM Intel-BEST 3.3 ONLY!  
20 REM LINE command demo  
30 DIM w1$(5),w2$(4),w3$(3)  
50 DATA ace,bat,cat,dog,eat  
60 DATA fat,got,hoe,ink  
70 DATA jar,key,let  
100 LINE 70: FOR x = 1 TO 3: READ w3$(x): NEXT  
110 LINE 60: FOR x = 1 TO 4: READ w2$(x): NEXT  
120 LINE 50: FOR x = 1 TO 5: READ w1$(x): NEXT  
200 IF w1$(1) = "ace" THEN PRINT " The new LINE command worked!": END  
210 PRINT "The LINE command failed???: END
```

THE NEW Ln8 COMMAND

SmartBASIC sets the TEXT window in both the graphics modes (GR and HGR) to four lines in length. Intel-BEST 3.3 includes a new command, 'Ln8', that will instantly increase the TEXT window to eight lines. To use this new command, just enter either graphics mode command and follow it with 'Ln8'. For example:

```
10 GR: Ln8
```

or,

```
10 HGR: Ln8
```

THE NEW PR#2 COMMAND

SmartBASIC includes the PR# command to transfer output to different devices. PR#1 transfers output to the ADAM printer. PR#0 transfers output to the video screen. Intel-BEST 3.3 includes a PR#2 command. If you have any parallel printer connected to ADAM via the external expansion port (on the right side of the memory console), this new command will transfer output to that device. As of the first printing of this software documentation, the EVE SP-1 and the EVE SP-1P are the only interfaces that connect to this external expansion port. If you use the PR#2 command with no device connected through this port, the results will be unpredictable.

THE COLOR TABLES ARE CORRECTED

Internally ADAM recognizes only one set of color codes. The GR and HGR codes are an arbitrary convention of SmartBASIC. Intel-BEST 3.3 corrects the COLOR, HCOLOR, and SCRN color tables. With this feature, you only have to learn (or refer to) one set of color values -- ADAM's master color code. These are:

- 0 = transparent
- 1 = black
- 2 = medium green
- 3 = light green
- 4 = dark blue
- 5 = medium blue
- 6 = dark red
- 7 = aqua/cyan
- 8 = medium red
- 9 = light red
- 10 = dark yellow
- 11 = light yellow
- 12 = dark green
- 13 = magenta
- 14 = gray
- 15 = white

Now you can use the same color values with COLOR, HCOLOR, and SCRNI! With just a little practice, you can even memorize the master color codes.

There are even more benefits with this feature. You can use transparent (the current background color) as a color. And you even have a wider range of colors to choose from as the standard SmartBASIC color tables omit some of the possible colors.

HOW TO CHANGE BACKGROUND COLORS

SmartBASIC V1.0 includes nine addresses that control screen colors. Three of these are for the background color. The remaining screen colors will appear to be on top of the background. When these other colors are set, the background appears as a border at the top and bottom of the screen.

Changing the background colors is very easy. Just select your color preference from ADAM's master color code table and POKE it into the appropriate address.

To change the TEXT background color:

POKE 17059, color: TEXT [RETURN]

To change the GR background color:

POKE 18607, color: GR [RETURN]

To change the HGR background color:

POKE 25431, color: HGR [RETURN]

HOW TO CHANGE THE NORMAL AND INVERSE COLORS

Changing the NORMAL and INVERSE colors is a little more complicated than changing the background colors. To make these techniques easier to understand, let's use the following variable names:

nl = NORMAL letters
ns = NORMAL screen
il = INVERSE letters
is = INVERSE screen
gw = graphics window

Just substitute your color code preferences for the variable names in the following equations.

TEXT NORMAL:

POKE 17115, (nl * 16) + ns: TEXT

TEXT INVERSE:

POKE 17126, (il * 16) + is: TEXT

GR window:

POKE 18633, (gw * 16) + gw: GR

GR TEXT:

POKE 18711, (nl * 16) + ns: GR

HGR window:

POKE 25471, (gw * 16) + gw: HGR

HGR TEXT:

POKE 25568, (nl * 16) + ns: HGR

Suppose you want the NORMAL TEXT letters to be white and the NORMAL TEXT screen to be dark blue. This is what you would enter:

POKE 17115, (15 * 16) + 4: TEXT [RETURN]

Transparent is colored with respect to the current background color. FLASH, in the TEXT mode, is the result of alternating between NORMAL and INVERSE colors. Be careful not to set corresponding letter and screen colors to the same value. If you do, you won't be able to read anything.

INSTANT SCREEN COLOR ROUTINES

Using any of the previously explained screen color changes has one drawback. You have to clear the screen in order to implement the color change. Intel-BEST 3.3 includes five machine language routines that will instantly change any of the screen colors without clearing the screen!

To facilitate the explanations, we'll employ the previously used variable names. Again, just substitute your color preference from ADAM's master color code for the variable names.

Instant background color change:
(works in TEXT, GR, or HGR mode)
POKE 1056, color: CALL 27472

Instant NORMAL TEXT color change:
POKE 1057, (nl * 16) + ns: CALL 27482

Instant INVERSE TEXT color change:
POKE 1058, (il * 16) + is: CALL 27489

Instant graphics mode TEXT color change:
(works ONLY in GR or HGR mode)
POKE 1059, (nl * 16) + ns: CALL 27503

Instant graphics window color change:
(works ONLY in GR or HGR mode)
POKE 1060, (gw * 16) + gw: CALL 27514

None of these routines affect the nine standard SmartBASIC addresses that control screen color. More explicitly, you can use the standard addresses as permanent control colors and use the Intel-BEST routines for quick impressive color changes. TEXT, GR, or HGR will reset the screen colors to the standard address values. PLEASE enter these POKE and CALL addresses VERY CAREFULLY!

MUSIC FUNDAMENTALS

Sound is caused by vibrating matter. The waves of these vibrations passing through the air beat against our eardrums and cause the sensation known as sound.

These sound waves arise from the condensation and rarefaction of air. The speed at which a sound wave travels is known as its frequency.

A musical sound is different from a noise. The difference lies in the fact that in a musical tone the frequency of vibration is uniform and regular. In noise the vibrations are irregular.

There are three fundamental characteristics of musical tones. 'Pitch' has to do entirely with the frequency of vibration of the sound waves -- the higher the frequency, the higher the pitch. 'Intensity' has to do entirely with the volume of the sound--the larger the amplitude of vibration, the louder the sound. 'Quality' is the characteristic that distinguishes one instrument from another. If you strike a middle-C note on a piano and play the same note with the same pitch and intensity on a violin, you can discern the different instruments because the quality is different.

A musical note is a symbol that represents a particular pitch or frequency. For example, if you strike the middle-C key on a piano, the ambient air is set in vibration 256 times per second. Thus, the middle-C note represents that particular frequency.

The natural notes are: A, B, C, D, E, F, and G. An A-sharp or a B-flat represents the frequency half way between the natural A and the natural B notes. This same principle applies to each of the flat and sharp notes.

An octave is the interval between one tone and another having twice as many or half as many vibrations per second. Thus, a C-note in the octave just below middle-C would have a frequency of 128 vibrations per second and the note would be lower. A C-note in the octave just above middle-C would have a frequency of 512 vibrations per second and the note would be higher.

The tempo is the overall speed of a group of notes. Duration is the relative length of time that a note is held. For example, you might decrease the tempo of a song (speed it up) and the duration of each particular note is likewise decreased. However, the notes will sound proportionally the same. A chord is a group of notes played in harmony (at the same time). For example, a 'C' chord is typically the C, E, and G notes played at the same time.

ADAM'S MUSIC FUNDAMENTALS

ADAM uses the Texas Instruments SN76489A sound chip. It has three square wave tone generators and a noise generator. Thus, it is capable of 4 simultaneous sounds: three independent voices and the noise sounds.

SmartBASIC (V1.0 and V2.0) provides no commands for accessing the sound chip. Without Intel-BEST 3.3, you can access the chip with machine language routines which transfer the data through any of the I/O ports numbered 224 through 255 (E0 - FF hex). Unless you are a machine language programmer, this method is rather complex.

Intel-BEST 3.3 gives you nine user-friendly commands that provide direct access to the sound chip. Each voice is capable of 1024 frequencies and the noise generator is equipped with 8 built-in sound effects.

Of the 1024 possible tones with each voice, only 48 correspond precisely to musical notes. The new Intel-BEST commands allow you to easily access these 48 notes.

THE NEW VOICE COMMANDS

Intel-BEST 3.3 includes three new commands for controlling the volume of each musical note (V1=, V2=, and V3=). It also includes three new commands that allow you to select the note for each voice (T1=, T2=, and T3=). Each of these commands is used in the same manner as COLOR=, HCOLOR=, and SPEED=.

The volume may be any integer between zero and fifteen. A zero will turn off the voice. Fifteen is the loudest setting. For example, if you want to turn on the volume for the first voice to the loudest setting, just do this:

V1 = 15 (then press [RETURN])

To turn it back off, enter this:

V1 = 0 (then press [RETURN])

This same principle applies to the volume for each voice. To change the volume for the second voice, use the new V2= command. To change the volume for the third voice, use the new V3= command.

The tone for each voice may be any value between zero and forty-seven. The following chart explains the note that corresponds to each tone value. The tones are separated into four groups. Each group represents one octave and contains twelve notes.

LOWEST OCTAVE:

VALUE	NOTE
0	A
1	A-sharp (B-flat)
2	B
3	C
4	C-sharp (D-flat)
5	D
6	D-sharp (E-flat)
7	E
8	F
9	F-sharp (G-flat)
10	G
11	G-sharp (A-flat)

MIDDLE OCTAVE:

VALUE	NOTE
12	A
13	A-sharp (B-flat)
14	B
15	C
16	C-sharp (D-flat)
17	D
18	D-sharp (E-flat)
19	E
20	F
21	F-sharp (G-flat)
22	G
23	G-sharp (A-flat)

HIGH OCTAVE:

VALUE	NOTE
24	A
25	A-sharp (B-flat)
26	B
27	C
28	C-sharp (D-flat)
29	D
30	D-sharp (E-flat)
31	E
32	F
33	F-sharp (G-flat)
34	G
35	G-sharp (A-flat)

HIGHEST OCTAVE:

VALUE	NOTE
36	A
37	A-sharp (B-flat)
38	B
39	C
40	C-sharp (D-flat)
41	D
42	D-sharp (E-flat)
43	E
44	F
45	F-sharp (G-flat)
46	G
47	G-sharp (A-flat)

To hear a low-C note, just do this:

T1 = 3: V1 = 15 (then press [RETURN])

To hear a middle-C note, just do this:

T1 = 15: V1 = 15 (then press [RETURN])

Just set the tone value equal to the note number that you want to hear. For the second voice, use the new T2= command. For the third voice, use the new T3= command.

THE NEW NOISE COMMANDS

Intel-BEST 3.3 includes two new commands for direct access to the noise generator (NV= and NS=). The NV= command may be set to any value between zero and fifteen. As with the voice volume commands, zero will shut off the noise and fifteen is the loudest setting. The NS= command allows you to select the noise that you want to hear. The following table explains each of the eight built-in sound effects.

NOISE SELECTIONS:

VALUE	NOISE
0	high pitched buzz
1	middle pitched buzz
2	low pitched buzz
3	helicopter noise
4	high motor hiss
5	low motor hiss
6	crash noise
7	erratic motor noise

SOUND COMMAND PROGRAMS

The following two pages LIST some simple demonstration programs using the new Intel-BEST sound commands. These programs as well as several others are stored on the Intel-BEST 3.3 datapak (or disk). Please study these LISTS.

Intel-BEST also includes another sound command, 'OFF'. This command will instantly shut off all voices and the noise generator. Thus, you can use the volume commands to turn the sounds off individually or you can use OFF to turn them all off simultaneously.

```

10 REM Intel-BEST 3.3 ONLY!
20 REM full scale demo
50 HOME: V1 = 0
100 PRINT: INPUT " What tempo please? ";tp%
110 PRINT: INPUT " What volume please? ";vo%
120 FOR note = 0 TO 47
130 T1 = note: V1 = vo%
140 FOR tempo = 1 TO tp%: NEXT tempo
150 NEXT note: OFF

```

```

10 REM Intel-BEST 3.3 ONLY!
20 REM natural notes of each octave
50 DIM nn%(7): HOME
60 DATA 0,2,3,5,7,8,10
70 FOR x = 1 TO 7: READ nn%(x): NEXT
80 DATA A,B,C,D,E,F,G
90 FOR x = 1 TO 7: READ nn$(x): NEXT
100 PRINT: PRINT: PRINT " LOW OCTAVE:": V1 = 0
110 FOR x = 1 TO 7: T1 = nn%(x): V1 = 15
120 GOSUB 600: NEXT x
200 PRINT: PRINT: PRINT " MIDDLE OCTAVE:": V1 = 0
210 FOR x = 1 TO 7: T1 = nn%(x)+12: V1 = 15
220 GOSUB 600: NEXT x
300 PRINT: PRINT: PRINT " HIGH OCTAVE:": V1 = 0
310 FOR x = 1 TO 7: T1 = nn%(x)+24: V1 = 15
320 GOSUB 600: NEXT x
400 PRINT: PRINT: PRINT " HIGHEST OCTAVE:": V1 = 0
410 FOR x = 1 TO 7: T1 = nn%(x)+36: V1 = 15
420 GOSUB 600: NEXT x: OFF
500 PRINT: PRINT: PRINT " THAT'S ALL FOLKS!": END
600 PRINT " ";nn$(x);
610 FOR tempo = 1 TO 375: NEXT tempo: RETURN

```

```

10 REM Intel-BEST 3.3 ONLY!
20 REM chord demo
50 V1 = 0: V2 = 0: V3 = 0
60 DATA 39,43,46
70 FOR x = 1 TO 3: READ cn%(x): NEXT
100 FOR x = 0 TO 3
110 T1 = cn%(1)-x*12: T2 = cn%(2)-x*12: T3 = cn%(3)-x*12
120 V1 = 13: V2 = 15: V3 = 15
130 FOR tempo = 1 TO 750: NEXT tempo
140 NEXT x: OFF

```

```
10 REM Intel-BEST 3.3 ONLY!
20 REM winner sound effects
50 V1 = 0: V2 = 0: total = 23
60 DIM note%(total),dur%(total)
100 DATA 34,2,26,2,28,1,26,1,23,1,26,2,23,1
110 DATA 21,2,23,1,21,1,20,2,21,1,20,1,17,2
120 DATA 20,1,21,1,20,1,21,1,23,2,26,2,34,2
130 DATA 26,2,26,3
200 FOR x = 1 TO total
210 READ note%(x),dur%(x)
220 POKE 1,128: POKE 2,note%(x)
230 POKE 4,164: POKE 5,note%(x)
240 V1 = 15: V2 = 15
250 FOR tempo = 1 TO dur%(x)*100: NEXT tempo
260 NEXT x: OFF
```

```
10 REM Intel-BEST 3.3 ONLY!
20 REM loser sound effects
50 V1 = 0: V2 = 0: total = 7
60 DIM note%(total),dur%(total)
100 DATA 35,1,37,1,35,1,33,3,35,6,28,3,26,4
200 FOR x = 1 TO total
210 READ note%(x),dur%(x)
220 POKE 1,128: POKE 2,note%(x)
230 POKE 4,164: POKE 5,note%(x)
240 V1 = 15: V2 = 15
250 FOR tempo = 1 TO dur%(x)*100: NEXT tempo
260 NEXT x: OFF
```

```
10 REM Intel-BEST 3.3 ONLY!
20 REM hodgepodge sound effects
100 DATA phone ring,sci-fi alarm,exit program
110 LINE 100: FOR x = 1 TO 3: READ opt$(x): NEXT
500 HOME: PRINT: PRINT " Please select an option:": PRINT
510 FOR x = 1 TO 3: PRINT " ";x;" = ";opt$(x): NEXT
520 GET key$: IF key$ < "1" OR key$ > "3" GOTO 520
530 key% = VAL(key$)
540 ON key% GOSUB 1000,2000,550: GOTO 520
550 TEXT: PRINT " That's all folks!": END
1000 FOR y = 1 TO 10: FOR x = 6 TO 15 STEP 3
1010 POKE 1,128: POKE 2,x
1020 FOR tempo = 1 TO 20: NEXT tempo
1030 V1 = 15: NEXT x: NEXT y: OFF: RETURN
2000 FOR y = 1 TO 3: FOR x = 1 TO 5: FOR z = 16 TO 2 STEP -1
2010 POKE 1,128: POKE 2,z+x
2020 POKE 4,160: POKE 5,x: POKE 7,192: POKE 8,z
2030 V1 = 13: V2 = 13: V3 = 13: NEXT: NEXT: NEXT
2040 OFF: RETURN
```

When working with the sound commands, you'll probably want to shut off the prompt bell. To do so, use the NB command explained on page 6 of this manual.

All six of the sound demonstration programs use a timing loop to control tempo and/or duration. Line 140 in the first program on page 18 is an example. A timing loop that counts from 1 to 750 lasts one second. A timing loop that counts from 1 to 1500 lasts two seconds, etc.

The three programs LISTed on page 18 are very easy to understand. Each one uses the Intel-BEST tone and volume commands. The three programs on page 19 access the sound chip by by-passing the tone commands. This technique warrants further discussion.

You can use any of ADAM's 1024 possible tones by POKEing in specific numbers at the appropriate addresses. To access a voice on the sound chip, ADAM requires that three values be transferred to the chip. These are: first frequency byte, second frequency byte, and volume byte. Intel-BEST 3.3 uses a machine code data table to store these values. When you use the tone or volume commands the values in this table are automatically changed. The following is a map of the Intel-BEST sound data table:

ADDRESS	SIGNIFICANCE
1	first voice, first frequency
2	first voice, second frequency
3	first voice, volume
4	second voice, first frequency
5	second voice, second frequency
6	second voice, volume
7	third voice, first frequency
8	third voice, second frequency
9	third voice, volume

By POKEing a value within the acceptable range (listed below) into a frequency address in the data table and then using an Intel-BEST volume command, you have complete access to all 1024 possible tones.

SOUND FREQUENCY RANGES:

first voice: first byte (128-143), second byte (0-63)
 second voice: first byte (160-175), second byte (0-63)
 third voice: first byte (192-207), second byte (0-63)

Using the Intel-BEST sound commands can add an exciting touch to your programs. If you are an experienced musician, you should be able to easily convert piano and organ sheet music to ADAM. Even if this is not your goal, playing with the sound commands can provide you with many hours of pleasurable entertainment.

TEXT BLOCK GRAPHICS

Intel-BEST 3.3 includes a machine language routine that instantly converts all the INVERSE letters to solid-colored blocks. To use it, just CALL 27561.

You can use these 8 by 8 pixel blocks right next to your TEXT. With a little ingenuity, you can create a variety of impressive displays. Please study the TBGdemo program on the Intel-BEST datapak (or disk). The TEXT command will reset the INVERSE letters.

THE BLOCK READ/WRITE ROUTINES

Intel-BEST includes three machine code routines for reading and/or writing to storage media blocks. Use these routines VERY CAREFULLY!

ADAM uses a special code to read or write to each storage device. The values are: tape one = 8, tape two = 24, disk one = 4, and disk two = 5. Rather than using SmartBASIC's drive suffixes when working with the OS commands, you can POKE any of these values into address 16821 to change devices.

These same values are used by the Intel-BEST media routines. Below is a list of these routines and their pertinent addresses. Again, use these routines VERY CAREFULLY. They are intended for the benefit of advanced BASIC programmers.

STANDARD BLOCK READ ROUTINE:

To use, CALL 27525
27526 = device code
27528 = block number to read from
27531 = low byte of RAM address buffer
27532 = high byte of RAM address buffer

STANDARD BLOCK WRITE ROUTINE:

To use, CALL 27549
27550 = device code
27552 = block number to write to
27555 = low byte of RAM address buffer
27556 = high byte of RAM address buffer

BLOCK READ ROUTINE WITHOUT INTERRUPT:

To use, CALL 27537
27538 = device code
27540 = block number to read from
27543 = low byte of RAM address buffer
27544 = high byte of RAM address buffer

The Intel-BEST storage medium contains a simple media copy program which demonstrates how to use the standard block read and write routines. Although the program is rather simple and easy to understand, you may find it to be a very useful utility for backing up your software.

Normally when ADAM reads from or writes to a storage block this operation is given sole priority. However, with the block read routine without interrupt a block can be read from while other activities are allowed to continue. SUPER GAME PACKS use a similar routine in order to let you play the game while it loads other parts of the game. You can use it in a similar manner to load shape tables and other short binary files without interrupting BASIC program operation.

This routine is default set to read the catalog on the tape in the first drive. In an integrated software package (programs RUN each other), this can be a great time saver, as ADAM has to read the catalog (block one) to determine each file's location and length.

ERROR MESSAGES

Out of Memory Error

You've tried to set LOMEM below 27600. Intel-BEST will not permit this.

Undefined Statement

You've referenced the LINE command to a non-existent line number.

Line Number Expected

You've inserted a colon or equal symbol between the LINE command and its line number or you forgot to use a line number.

Illegal Quantity Error

You've used a value with one of the sound commands that's not within the acceptable range.

Illegal Variable Name

You've tried to use a BASIC or Intel-BEST command or an Intel-BEST shortcut command as the name for a variable.

Meaning Of Line Unclear

You've tried to use a BASIC or Intel-BEST command or an Intel-BEST shortcut command as the name for a variable.

HOW TO COPY Intel-BEST TO A BASIC BACKUP

Of course, you can use a media copy utility to backup Intel-BEST 3.3. However, you may want to copy the program to a SmartBASIC backup for single medium loading. Here's how:

TO COPY HELLO:

1. NEW
2. LOAD HELLO (from Intel-BEST 3.3)
3. SAVE HELLO (to SmartBASIC V1.0 backup)

TO COPY Intel-BEST 3.3:

1. NEW
 2. LOMEM:30000
 3. BLOAD Intel-BEST,A27600
 4. BSAVE Intel-BEST,A27600,L1742 (to SmartBASIC V1.0 backup)
 5. LOMEM:27600
-

WHAT ABOUT COMPATIBILITY

You can still RUN most SmartBASIC V1.0 programs after Intel-BEST 3.3 is active. However, the reverse is not true. Thus, Intel-BEST is upwardly compatible with SmartBASIC. Intel-BEST is not compatible with SmartBEST V1.0 which it is patterned after.

Only three of the documented SmartBASIC V1.0 commands are replaced by Intel-BEST 3.3. These are LET, TRACE, and NOTRACE. Of all SmartBASIC's 100-plus commands, these three are the least used. Of course, you can still use the understood LET command, ie, not spelling out the word LET. To use the TRACE function, just CALL 6336. To use the NOTRACE function, just CALL 6341.

If you have a problem RUNing a SmartBASIC V1.0 program with Intel-BEST, it probably uses one of the new commands as a variable name. In this event, BASIC will simply omit the program line containing the illegal variable name. The easiest way to correct the SmartBASIC program is to use SmartWriter's SEARCH and REPLACE functions to change the illegal variable names.
