

2012-04-07

ABasiC errata and index pictures taken by markk@clara.co.uk

The pictures consist of:

- One errata sheet (two sides). That might have been supplied with the ABasiC manual. It is probably earlier than the rest of the pages.
- Other sheets with more errata, some extra pages for the manual and an index. Those were probably sent by Commodore to registered users.

Page No.	Description
R-142	Addition of the CHDIR command description. Insert the following after the CHAIN MERGE command description:

CHDIR CHDIR "file-spec"

Use the CHDIR (CHange DIRectory) command to change the current directory. When you enter ABasiC, execute a DIR command, or LOAD an ABasiC program using only a file name (that is, without a volume or drive number), the system automatically uses the main, or root, directory.

For example, if you want to store a set of programs under a special subdirectory, you can save, load, and run those programs in two ways: The first way is to specify the subdirectory and/or the drive number before the file name each time you perform one of the system commands that requires this information. The second way is to execute a CHDIR command to change the directory to the volume name and subdirectory name. Then simply specify the file name when you wish to load, save, or run a particular program.

```
CHDIR "APROGRAMS:MISC"  
SAVE PROG42
```

The above commands change the current directory to subdirectory MISC on the disk with volume name APROGRAMS. You can perform the same job with the following command:

```
SAVE APROGRAMS:MISC/PROG42
```

You can use a similar format for the other system commands that involve a file specification, such as LOAD, REPLACE, RUN, and so forth.

The CHDIR command also affects other types of file activity, such as operations on data files. For example, instead of the following statement:


```
OPEN "O", #1, "APROGRAMS:MISC/PROG42DATA"
```

you can change the directory:

```
CHDIR "APROGRAMS:MISC"
```

and then use the following OPEN command:

```
OPEN "O", #1, "PROG42DATA"
```

Note that the DIR and SHELL commands operate on the root (that is, the highest level) directory only. They are unaffected by a CHDIR command executed in ABasiC. See the AmigaDOS Users Manual for further information on file specification.

- R-169 Addition of information on file specification with certain system commands. Add the following text after the last paragraph on the introduction to the system commands:

Note: If you use a system with one disk drive, you must specify

```
DF0:<file name>
```

instead of the simple file name to instruct ABasiC to use the disk currently in the drive. Without the drive specification, "DF0:", the operating system will request that you put the WorkBench disk in the drive before it executes the system command you entered.

Similarly, if the file you want is on a particular volume or subdirectory, you must specify this information before entering the file name. For example, to RUN a disk file named "APROGRAM" from the disk with volume name "JUNK", use the following statement:

```
RUN JUNK:APROGRAM
```

If the specified volume is not in the drive, the system will request that you insert it before executing the above statement.

Attached is a set of errata pages for your *ABasiC* manual.

Instructions are provided in the errata to indicate how the appropriate manual pages should be modified.

You may choose to make these changes or simply to file this errata package with the manual pages for future reference.

Commodore-Amiga, Inc.

Itemization of Errata

Page No.	Description
R-34	Correction of text explaining use of array values with SWAP command: Last line should read, "You can SWAP array values, but not the arrays themselves."
R-48	Replacement of format string for the minus sign, top of page. Should be: "##.##-"
R-55	Addition of semicolon at end of line 10 in POS programming example; affects the value returned.
R-73	Revision of code in WHILE programming example to produce desired output: switch lines 10 and 20
R-84	Correction of text in explanation of TAN function: tangent is sine over cosine; also, switch "SINE" and "cos" in diagram illustrating relationships
R-85	Correction of programming example, line 50, and revision of output from PI, which is a <i>double</i> number. Change line 50: 50 PRINT (ATN(.5774))*(180/PI) and revise output: 8660 5000 57735 30.0021386840363
R-87	Replacement of semicolons by plus signs in line 10 of CHR\$ programming example: 10 A\$="STAIR"+CHR\$(10)+SPACE\$(5)+"STEP"
R-96	Note that the ANIMATE command might not be currently implemented
R-98	Replacement of semicolon by colon after PENB statement, line 10 of programming example

- R-100 Correction of the BOX command description. Replace last sentence in first paragraph with the following sentence:
- "Otherwise, BOX uses the current pixel cursor coordinates."
- R-104 Correction of description of mode 1 for DRAWMODE, top paragraph. Replace phrase, "(or the current output window)" with "at the drawing point".
- Correction of PENA, PENB, and FONT syntax in lines 10 and 20 of programming example. Replace lines 10-20:
- 10 PENA 2: PENB 4
20 FONT 1: DRAWMODE 0
- R-105 Addition of warning that FONT clears the screen. Add the following sentence to end of FONT description:
- "Note: FONT clears the screen or current window and thus should execute before any graphics commands in your program."
- R-121 Revision of text explaining calculation of array size for SSHAPE command; the second paragraph should read:
- "Divide the width by 16 and round up to the next integer. Multiply the result by the height and the display depth (the default depth is 4), and add 4 to the product. Divide this number by 2 to get the number of long words--that is, the number of cells in the integer array. The result is the number you should use to DIMension the array you're saving the rectangle in."
- R-128 Replacement of text string in line 20 of programming example for the NARRATE command: the string should be all capital letters.
- R-130 Correction of the second equation on the page: switch the numerator and denominator
- R-133 Replacement of -1 with 0 for third parameter of SOUND statement in line 130 of SOUND programming example; addition of "x% =" at beginning of same line for correct syntax
- R-134 Addition of parentheses around quoted string following TRANSLATES command, line 30 of programming example

- 16002
64008
- R-138 Explanation of lowering pitch by the doubling of the waveform length;
Add second paragraph:

"Below is a table containing the period values for the standard musical scale. To produce a tone one octave lower than the one you obtain with the values in this table, double the length of the waveform table you're using (see the WAVE command). To produce a higher pitch, reduce the waveform table by half."

- R-139 Emphasis that ABasiC treats peripherals and windows as files; add the following sentence to the second paragraph:

"Note that ABasiC considers peripherals (printers, plotters, etc.) and custom windows as files, and thus you must open, send output to, and close them as you would a disk file."

- R-140 Addition of quotes around file name in line 20 of programming example for BLOAD

- R-141 Addition of quotes around file name in programming example for BSAVE

- R-148 Correction of LIBCALL programming example and addition of explanation, top of page. Add following text to LIBCALL explanation, second paragraph:

"You must close the library in a similar manner--ABasiC does not close it automatically. Note: The offset (-408 in the following example) is subject to change. See the Amiga ROM Kernal Manual for a list of the library functions and their offsets."

and use the following replacement lines in the LIBCALL programming example:

```
100 LIBNAM$= "graphics.library" + CHR$(0)
120 ARGS$(9)=PEEK_L(VARPTR(LIBNAM$)+2):' pass to reg. A1
```

Replacement of "PRINTER" with "PRT:" in first text example for OPEN command, bottom of page.

- R-152 Replacement of 2 with 0 as first argument of SCREEN examples; in text and in line 50 of SCREEN programming example

- R-153 Correction of explanation of SHELL command. Delete all of paragraph 1 except first sentence.
- Replacement of "LP" with "PRT:" in SHELL programming example
- R-155 Replacement of 2 with 1 as first argument of SCREEN example and in line 10 of WINDOW programming example
- R-160 Addition of comma after "I" in line 80 of LINE INPUT# programming example
- R-167 Correction of text for LOC description; replacement of GET and PUT by RGET and RPUT, respectively
- R-172 Replacement of "printer" with "prt:" in text (last paragraph) and programming examples
- R- 175 Addition of requirement of using extensions on file names with the RENAME command. Add the following sentences to command description:
- "Amiga DOS automatically attaches the extension ".bas" to an ABasiC program you store. Be sure to specify this extension (or any other) when you RENAME a file".
- A-15 Deletion of Amiga DOS Error Codes, pp. A-15 and A-16 from Appendix B
- A-17 Deletion of phrase, "(or Enter)" from description of ASCII code 3
- A-18 Deletion of phrases, "(or [up arrow])", etc. from description of ASCII codes 65 through 68
- A-27 Addition of Phoneme Table (pp. A-27 and A-28) at end of Appendix D(attached)
- Addition of Index (attached)

Note: If you attach an external disk drive to the Amiga, it uses about 24k of memory. Thus, you should reduce the display depth (see the SCREEN command) to free additional memory.

Using Punctuation

In addition to periods and question marks, Narrator recognizes the dash, comma, and parentheses. The comma goes where you would normally put it in a written English sentence and tells Narrator to pause with a slightly rising pitch, indicating that there is more to come. For example, you may find that you can add more commas than you use in written English to help set off clauses from each other.

The dash is like the comma except that the pitch does not rise so severely. Here's a rule of thumb: Use dashes to divide phrases and commas to divide clauses. Parentheses provide additional information to Narrator's intonation routine. Put them around noun phrases of two or more content words, for example "giant yacht." Parentheses can be particularly effective around large noun phrases like "the silliest guy I ever saw." They help provide a natural contour.

Hints for Intelligibility

Although this guide should get you off to a good start, the only sure way to proficiency is to practice. Follow these tricks to improve the intelligibility of a sentence:

1. Polysyllabic words are often more recognizable than monosyllabic ones. So say enormous instead of huge. The longer version contains information in every syllable and gives the listener three times the chance to hear it correctly.
2. Keep sentences to an optimal length. Write for speaking rather than for reading. Do not write a sentence that cannot be easily spoken in one breath. Keep sentences confined to one idea.
3. Stress new terms highly the first time they are heard.

These techniques are but a few of the ways to enhance the performance of Narrator. Undoubtedly, you'll find some of your own. Have fun.

Table of Phonemes

Vowels

Phoneme	Example	Phoneme	Example
IY	beet	IH	bit
EH	bet	AE	bat
AA	hot	AH	under
AO	talk	UH	look
ER	bird	OH	border
AX	about	IX	solid

AX and IX should never be used in stressed syllables

Diphthongs

Phoneme	Example	Phoneme	Example
EY	made	AY	hide
OY	boil	AW	power
OW	elow	UW	cre

Consonants

Phoneme	Example	Phoneme	Example
R	red	L	yellow
W	away	Y	yellow
M	men	N	men
NX	sing	SH	rush
S	sail	TH	thin
F	fed	ZH	pleasure
Z	has	DH	then

Phoneme	Example
V	very
CH	check
/H	hole
B	but
D	dog
K	Commodore

Phoneme	Example
J	judge
/C	loch
P	put
T	toy
G	guest

Special Symbols

Phoneme	Example	Phoneme	Example
DX	pity (tongue flap)	Q	kitt_en (glottal stop)
QX	pause (silent vowel)		
RX	car (postvocalic R and L)	LX	call
UL	=AXL	IL	=IXL
UM	=AXM	IM	=IXM
UN	=AXN	IN	=IXN

(contractions—see text)

Digits 1–9	syllabic stress, ranging from secondary through emphatic
.	period—sentence final character
?	question mark—sentence final character
–	dash—phrase delimiter
,	comma—clause delimiter
()	parentheses—noun phrase delimiters (see text)

Index

+ R-11, R-47
- R-11, R-48
_ R-50
* R-11
/ R-11
\ R-11
^ R-11, R-50
** R-11, R-49
= R-22, R-28
< R-22
> R-22
<= R-22
>= R-22
< > R-22
' R-33
: R-3, R-42
; R-43
, R-43, R-48
. R-48
? R-44
! R-51
R-47, R-143
& R-52
\$\$ R-49
**\$ R-50
\\ R-52
[blank] R-3, R-5
absolute value R-74
ABS R-74
address R-82
ALL R-141
AND R-23
ANIMATE R-96
APPEND R-156
AREA R-96
argument R-2
Arithmetic Functions R-74
arithmetic operator R-10
array R-10, R-29
ASC R-86
ASK CURSOR R-98
ASK MOUSE R-98
ASK RGB R-99
ASK WINDOW R-139, R-100
aspect ratio R-101
assembly language routine R-148
Assignment Commands R-28
AT() R-42, R-44
ATN R-84
AUDIO R-123
AUTO R-169
BASIC 2
binary operation R-11
BLOAD R-140
Boolean operator R-26
BOX R-100
branch R-62, R-68
BREAK R-179
BSAVE R-140
bug R-179
byte R-149
CDBL R-75
CHAIN R-141
CHAIN MERGE R-142
channel mask R-123
character coordinates R-95

CHR\$ R-87
 CINT R-76
 CIRCLE R-101
 CLOSE R-157
 CLR R-28
 CMD R-143
 color clocks R-129
 color registers R-118,
 R-122
 command R-2
 COMMON R-145
 concatenate R-13
 constants R-4
 CONT R-170
 COS R-84
 counter variable R-58
 CSNG R-75
 Ctrl-C R-61, R-170
 Ctrl-D R-171, R-179
 CVD R-166
 CVI R-166
 CVS R-166
 DATA R-35
 Data File Commands R-156
 Debugging Commands R-179
 DEC R-76
 declare R-8
 DEF FN R-146
 deferred mode R-4
 DELETE R-170
 delimiter R-3, R-14
 depth (display) R-95, R-152
 Derived Trigonometric
 Functions R-85
 device R-148
 DIM R-29
 DIR R-147
 disk file R-139

double-precision R-6
 DRAW R-102
 DRAWMODE R-103
 e (exponential) R-76
 EDIT R-18, R-171
 element R-2
 ELSE R-62
 END R-57
 envelope R-135
 EOF R-167
 EQV R-24
 ERASE R-31
 ERL R-67, R-171,
 R-180
 ERR R-67, R-180
 ERR\$ R-182
 ERROR R-180
 error trap R-69, R-181
 EXECBASE R-148
 execute R-4
 EXP R-76
 exponent R-12
 expression R-2
 FIELD R-157
 field R-35, R-157
 file buffer number R-143
 file header R-141
 File Input/Output Functions R-166
 File Management Commands R-139
 file marker R-156
 file mode R-161
 file specification R-147
 FIX R-77
 flag R-21
 flood fill R-113
 FOLLOW R-182
 FONT R-105
 FOR R-58

FN R-146
 FRE R-77
 Functions R-74
 fundamental R-137
 GET R-36
 GET # R-158
 GETKEY R-37
 GOSUB R-59
 GOTO R-61
 GRAPHIC R-38
 Graphics Commands R-95
 Graphics Functions R-121
 GSHAPE R-105
 hardware register R-147
 HELP R-171, R-183
 HEX\$ R-87
 hexadecimal R-5, R-87
 IF R-62
 immediate mode R-4
 IMP R-24
 increment R-59
 indefinite loop R-72
 index R-147
 initialize R-8
 INPLAY R-137
 INPUT R-39
 input R-35
 Input/Output Commands R-35
 INPUT # R-159, R-41
 INSTR R-88
 INT R-78
 integers R-4
 INVERSE() R-42, R-45
 inverse video R-45
 keyword R-2
 LEFT\$ R-88
 LEN R-89
 LET R-32

LIBCALL R-147
 library functions R-147
 LINE INPUT R-41
 LINE INPUT # R-160, R-41
 line number R-4
 LINEPAT R-107
 LIST R-171
 LIST # R-171
 LOAD R-173
 LOC R-167
 LOCATE R-108
 LOF R-167
 LOG R-79
 LOG10 R-79
 logical line R-2
 Logical Operators R-23
 long-word R-149
 loop R-58, R-72
 LSET R-160
 MAT AREA R-109
 MAT DRAW R-110
 matrix R-29
 MERGE R-173, R-142
 MID\$ R-90
 MKD\$ R-168
 MKI\$ R-168
 MKS\$ R-168
 MOD R-80
 modulo R-80
 NARRATE R-124
 nested loop R-60, R-73
 NEW R-174
 NEXT R-64, R-70
 NOT R-25
 null string R-7
 numeric variable R-8
 OCT\$ R-91
 octal R-5, R-91

ON ERROR GOSUB R-66
 ON ERROR GOTO R-66
 ON GOSUB R-67
 ON GOTO R-68
 OPEN R-148, R-161
 operand R-12
 operator R-2
 Operators R-21
 OPTION BASE R-32
 OR R-25
 OUTLINE R-97, R-110
 outline pen R-95
 output R-35
 overlay R-142
 overtone R-137
 PAINT R-113
 parameter R-2
 PATTERN R-115
 PEEK R-149
 PEEK_L R-149
 PEEK_W R-149
 PENA R-118
 PENB R-119
 PENO R-119
 PERIOD R-128
 phonemes R-124
 PI R-83
 PIXEL R-121
 pixel coordinates R-95
 pointer R-147
 POKE R-149
 POKE_L R-149
 POKE_W R-149
 POS R-55
 precedence R-14
 primary pen R-95
 PRINT R-42
 PRINT # R-162

PRINT AT() R-42, R-44
 PRINT INVERSE() R-42, R-45
 PRINT USING R-42, R-46
 PRINT # USING R-163
 Program Flow Commands R-57
 program mode R-4
 prompt R-37, R-40
 queuing R-133
 radius R-101
 random access file R-156
 RANDOMIZE R-32
 READ R-52
 record length R-162
 recursion R-146
 Relational Operators R-21
 relationship R-21
 REM (also ') R-33
 RENAME R-175
 RENUMBER R-175
 REPLACE R-177
 REPLACES R-33
 reserved word R-2
 resolution R-152
 RESTORE R-53
 RESUME R-69
 RETURN R-70
 RGB R-119
 RGET # R-163
 RIGHTS R-91
 right-justify R-165
 RND R-80
 RPUT # R-164
 RSET R-165
 RUN R-177
 sampled sound R-128, R-136
 sampling period R-128
 SAVE R-178
 scientific notation R-6

SCNCLR R-54
 SCRATCH R-151
 SCREEN R-151
 screen R-151
 Screen Position Functions R-55
 secondary pen R-95
 seed R-33
 sequential file R-156
 SGN R-81
 SHELL R-153
 SIN R-84
 single-precision R-6
 SLEEP R-153
 SOUND R-131
 Sound Functions R-137
 SPACES\$ R-91
 SPC R-55
 Speech & Sound
 Commands R-123
 SQR R-82
 square root R-82
 SSHAPE R-120
 statement R-3
 STATUS R-134
 STEP R-58
 STOP R-70
 STR\$ R-92
 STRING\$ R-92
 string constant R-7
 String Functions R-86
 string variable R-10
 subroutine R-59
 substring R-88
 SWAP R-34
 syntax R-3
 syntax conventions R-15
 syntax error R-179
 SYSTEM R-154
 System Commands R-169
 TAB R-56
 TAN R-84
 THEN R-62
 TO R-58, R-96,
 R-102, R-175
 TRACE R-183
 TRANSLATES R-134
 Trigonometric Functions R-83
 TROFF R-184
 TRON R-184
 unary minus R-13
 UNBREAK R-185
 UNFOLLOW R-185
 UNTRACE R-185
 USING R-42, R-46
 VAL R-93
 variables R-8
 VARPTR R-82
 VOLUME R-135
 WAVE R-136
 waveform R-135
 WEND R-72
 WHILE R-72
 WIDTH R-54
 WINDOW R-154
 word R-149
 WRITE # R-165
 XOR R-26

