PRODUCT UPDATE

ATARI HOME COMPUTER SYSTEM

ATARI® Assembler Editor User's Manual Update

This product update contains a number of corrections and additions to the *ATARI Assembler Editor User's Manual* (C014189-03 REV. 1).

Inside Front Cover. The codes listed are **BASIC Error Codes,** and should be replaced with **Assembler Error Codes** as shown (which are also listed in Appendix 1):

Error No.	
1	Insufficient memory
2	No start address for DEL command
3	Mini-assembler address error
4	LOAD file error
5	Undefined label reference
6	Error in syntax of statement
7	Label defined more than once
8	Buffer overflow
9	Missing label
10	Value greater than 255
11	Invalid null string
12	Incorrect address or address type
13	Phase error
14	Undefined forward reference
15	Line too large
16	Unrecognizable source statement
17	Line number too large
18	Misuse of LOMEM command
19	No starting address
20	Overflow in NUM or RENUM
128	BREAK key pressed during I/O operation
130	Nonexistent device
132	Invalid command
136	End of file
137	Record longer than 256 characters
138	Device does not respond
139	Device does not return Acknowledge signal
140	Serial bus input framing error
142	Serial bus data frame overrun
143	Serial data checksum error
144	Device done error
145	Read-after-write compare error
146	Function not implemented
162	Disk full
165	Filename error

Page vii. These are the correct page numbers for:

ATARI®

Page viii. In the Appendices, the title for **Appendix 9** should read:

9 **Using the Assembler Editor Cartridge to Best Advantage: Sample Programs** 63

Page 1 and **Page 2.**

Delete reference to ATARI 815 Dual Disk Drive.

Page 5. The caption for **Figure 2** should read:

Figure 2. Memory map without use of LOMEM.

Page 9.

The example shown in Figure 4 is not an executable program. It is used only to demonstrate the format of a sample program.

Page 11. The third paragraph should read:

Please refer to the description of the LABEL= directive . . .

Page 12. This information pertains to **Indirect Indexed Operands:**

Using indirect indexed operands will sometimes produce an error 12, although the source code appears to assemble correctly anyway. Use with caution; examine the object code to be certain.

Page 17. Under the **REP Command**, the first listing in the left column should read:

REP/OLD/NEW/

Page 18. On the sample **Programming Form**, the following line numbers are missing:

20
30
40
50
60
70
80
90

In **Figure 7** on line 50, IMY should be:

INY

Page 19. This information pertains to the **LIST Command:**

The LIST command does not set the display flag, so a LIST containing control characters will execute those functions instead of printing the characters.

The LIST Command format should read:

$$\text{LIST}\left[\left\{\begin{array}{l} xx[,yy] \\ \#\left\{\begin{array}{l}\text{device:}\\ \text{filespec}\end{array}\right\}[,xx[,yy]] \end{array}\right\}\right]$$

Page 20. The program should read as follows—note in particular the indentation and spacing:

```
EDIT
LIST RETURN
10   *= $3000
20   LDY #00
30 REP LDX ABSX, Y
40   BNE XEQ SAME PAGE
50   INY TALLY
60   JMP REP
70 ABSX = $3744
80 XEQ = *+$60
90   .END

EDIT
LIST30 RETURN

30 REP LDX ABSX, Y

EDIT
LIST 60,80 RETURN

60   JMP REP
70 ABSX = $3744
80 XEQ = *+$60

EDIT
```

Page 22. The **SAVE Command Example** should look like this:

Examples: SAVE#C:<1235,1736
SAVE#D2:MYFILE<1235,1736
SAVE#C9:<1235,1736

To save an object program residing in hex address1 to address2 on cassette or diskette, the commands are:

SAVE#C: < address1,address2
CAUTION: Use the CSAVE procedure illustrated in your 410 Program Recorder Operator's Manual.

SAVE#D:FILENAME < address1,address2
where FILENAME is an arbitrary name you give to the block of memory that you are saving (where your object program is stored).

SAVE#C9: <address1,address2
Saves an object file without the DOS header bytes. This command procedure is used to generate tapes that can be booted on the cassette. You will have to supply your own cassette boot control information in your program. (See section 10 of Technical Reference Notes CA016555)

Page 22. The **LOAD Command Format** should look like this:

Format: LOAD# { device: / filespec }

Page 25. In the general form diagram the upper left line should read:

ASM [#D[n]:PROGNAME[.SRC]]

Add to Figure 8.

In Figure 8, an additional line will appear at the end of the assembly, prior to the next EDIT prompt. The added line shows the number of errors in the assembly.

Page 26. This information pertains to where object program is to be stored:

When generating tapes that can be booted on the cassette, do not assemble directly to the cassette using ASM,,#C9:. Store the object code in RAM and then use SAVE#C9:< address 1, address 2.

Page 28.

One more space should be added between the line numbers and pseudo ops on the top half of the page.

In addition, one O should be deleted from the sixth line down so that it reads:

100 .OPT NOOBJ

Under **Title and Page Directives,** add the following note:

The number of lines per page is set at 56 lines. To change the number of lines per page, change the content of location 480H. To cause an assembly listing to feed to top of page, change the location 481H to a non-zero value.

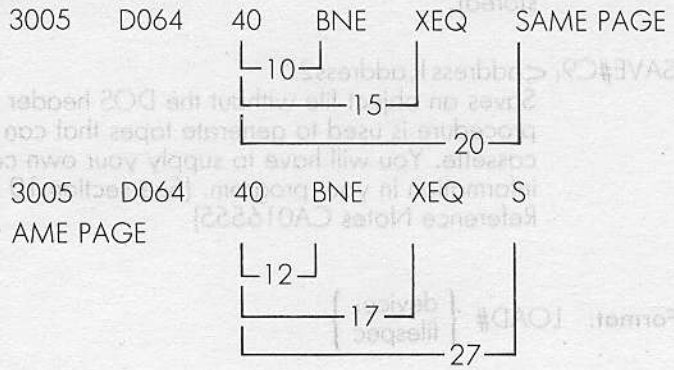
For example:
BUG RETURN
DEBUG
C480<3C
(this changes the number of lines in the printout to 60)

Page 29. Under the **Tab Directive,** the following clause and table should be inserted after the form of the directive (third paragraph):

—where each number corresponds to the number of columns to the right of the line number field to start printing.

Number	Field
1	op code
2	operator
3	comment

The last two drawings at the bottom of the page, should be replaced by:



ATARI®

Page 30. Under **BYTE, DBYTE and WORD Directives**, the second line should read:

```
200     .BYTE "AB. . . N"
```

In the third paragraph of text under **BYTE Directive**, the last line should read:

Appendix 5.

Page 31. The first paragraph of text under **LABEL = DIRECTIVE**, the second line should read:

the sample program we used before. Statements 70 and 80 give. . . .

Under **LABEL = DIRECTIVE**, lines 60 and 70 in the program should read:

```
70 ABSX = $3744
80 XEQ = *+$60
```

In the following paragraph, the last line should read:

given in Appendix 5.

This note belongs just above * = **Directive:**

Note: The assembler will always assign two-byte values to forward-referenced labels. To use zero page addressing correctly, be certain that all zero page labels are defined ahead of time.

Page 32. The fifth paragraph of text, first line, should read:

The effect of the directive is to reserve 36 locations immediately. . . .

Under **IF Directive**, the first paragraph of text, last sentence, should read:

If the expression is not equal to zero, all of the code between lines 900 and 990 will not be assembled.

The **SOURCE CODE** program should be rewritten as follows—note in particular the spacing requirements:

```
0100  ;CONDITIONAL ASSEMBLY EXAMPLE
0120  Z =0
0130    *= $5000
0140    LDA $45
0150    . IF Z @ZNOTEQUAL0
0160    TAX ;THIS CODE ASSEMBLED IF Z=0
0170  ZNOTEQUAL0
0180    . IF Z-1 @ZNOTEQUAL1
0190    ASL A ;THIS CODE ASSEMBLED IF Z=1
0200  ZNOTEQUAL1
0210    INX ;THIS CODE ALWAYS ASSEMBLED
```

ATARI®

Page 33. Last line of the listing.

An additional line will appear. The line will be the last line of each line of each assembly listing, and it will indicate the number of errors that were detected during the assembly.

Page 35. Under **DEBUG COMMANDS**, fifth line from the bottom, the command for Trace Operation should read:

T or Tmmmm

Page 36. The information for the **D or Dmmmm Display Memory** command should read:

Dmmmm, yyyy where yyyy is less than or equal to mmmm shows the contents from mmmm to yyyy, inclusive, with address "wraparound" occurring at address $FFFF.

The **Example** should be changed to read as follows:

DFFF0,3 RETURN

FFF0	68	40	FF	FF	FF	FF	FF	FF
FFF8	DD	57	B4	E7	77	E4	F3	E6
0000	00	41	41	41				

Page 37.

The top sentence, beginning "This shows that address 5000. . . .," should be deleted.

In the following example, the fourth line should read:

5008	18	41	54	41	52	49	20	20

Page 38. The first sentence should read:

The second command puts 31 and 87 in locations 700B and 700E. . . .

Under **Vmmmm Verify Memory**, the first sentence should read:

Vmmmm <yyyy,zzzz compares memory yyyy to zzzz with. . . .

Page 40. The **Example** should be changed as shown—note in particular the spacing requirements:

```
5001< LDY  $1234  RETURN
5001   AC3412  Computer Responds.
< INY  RETURN
5004   C8  Computer Responds.
```

This note should follow the **Example**:

Note: Always leave a space after the < when specifying a mnemonic to be assembled under this command.

This information pertains to **Gmmmm Go (Execute Program)**:

A BRK (op code = $00) instruction will also stop the GO command. So even though the debugger does not support an explicit breakpoint facility, you can simulate this facility with a little extra work.

Page 41. This information pertains to the TRACE operation:	When creating a program for tape that can be booted on the cassette, the execution of the program cannot be traced if the cassette boot control information header has been included in the program. The program should be debugged before adding the header.
Page 47. This definition should immediately follow the first paragraph:	' The apostrophe indicates that the following character is to be translated into ASCII code.
In the program line following the asterisk (*) definition, one space should be deleted between the number 50 and the word HERE so that it reads:	50 HERE = * + 5
In the last paragraph of text on the page, the first sentence should read:	The asterisk also signifies multiplication (see Appendix 5).
Page 49. In the last paragraph, left column under the table, the title should read:	**Z, PAGE X-Z, PAGE Y-ZERO PAGE INDEXED**
In the first paragraph in the right column, the title should read:	**ABS, X-ABS, Y-ABSOLUTE INDEXED**
In the last paragraph in the right column, the title should read:	**(IND), Y-INDIRECT INDEXED**
Page 51. In the example at the bottom of the page, line 600 should read:	600 LDA #LABEL&$00FF
Line 620 should read:	620 LDA #LABEL /256

ATARI®

Page 53. In the first column, the third directive, should read:

.PAGE ''MESSAGE''

In the first column, the fifth directive, should read:

.BYTE a,b, . . . , n

In the first column, the seventh directive, should read:

.DBYTE a, b, . . . , n

The second directive from the bottom should read:

assembles following code, up to LABEL, if only . . . .

Page 60.

The **Notes** at the bottom of the page refer to ATARI BASIC and do not pertain to Assembly language.

Page 61. These are additional ATARI PUBLICATIONS:

ATARI Home Computer System Technical Reference Notes CA016555 Rev. A
DOS II Utility Listings C017894

Delete reference to:

ATARI 815™ Operator's Manual C016377

This is an addition to OTHER PUBLICATIONS:

The Atari Assembler by Don and Kurt Inman, Reston Publishing Co., Reston, VA

Page 63. The Appendix title should read:

Using the Assembler Editor Cartridge To Best Advantage: Sample Programs

Page 64. This information pertains to using a Program Recorder to assemble a program:

Do not assemble directly to tape (ASM, , #C:) except with very short programs. A longer program will result in a timeout error. Instead, assemble in memory (ASM) and save to tape (SAVE#C:<startadr,endadr).

Page 65. The second sentence in the first paragraph should read:
And this is the program that follows:

If you have a cassette, type in the following program for loading from cassette tape:

```
100  TRAP 300
110  OPEN #1, 4, 0, "C:":REM open file on cassette for input
120  GET #1, X:GET #1, X:REM throw away first two bytes
130  GET #1, LOBYTE: GET #1, HIBYTE
140  START=HIBYTE*256+LOBYTE:REM starting address
150  GET #1, LOBYTE: GET #1, HIBYTE
160  FINISH=HIBYTE*256+LOBYTE: REM ending address
170  FOR ADDRESS=START TO FINISH
180  GET #1, BYTE
190  POKE ADDRESS, BYTE
200  NEXT ADDRESS
210  GOTO 130: REM check for further sections of code
300  CLOSE #1
```

Then a new paragraph starts at:

When the machine language. . . .

Page 67. Line 25010 should read:

```
25010?J+5;"E$(";A;",";B;")=";CHR$(34);
```

Line 25020 should read:

25020 FOR I=A TO B:?"ESC ESC";CHR$(PEEK(I+C));:NEXT I

Pages 68-74.

In the programs listed on page 68-74, the spacing is wrong, there should be only one space between the third and fourth columns, one space between the fourth and fifth columns, and one space between the fifth and sixth columns.
At the end of each listing, an additional line that indicated zero errors will appear for each successful assembly.

Page 69. The information in lines 0250 and 0260 is transposed; the lines should read:

```
0250          PLA
0260          STA      ATTACK      SET ATTACK TIME
```

In line 0420, the O in #$BO should be a zero.

In line 0470, the O in #$OE should be a zero.

Page 70. The third line in **Example 3** should read:

```
30          ; ROUTINE SPLAY
```

In line 0180, the last O in COLORO should be a zero.

Page 71. Lines 0630 and 0640 should read:

```
630   ;
640   ; BITS 3 AND 4 NOW GIVE THE COLOR TO USE
```

Page 72. In Example 4, lines 10 and 20 should read:

```
10    ;
20    ; KATHY'S COLOR PALETTE
```

Page 73. Line 400 should read:

```
0400   LOOP1   STA   (POINTA),Y
```

Page 74. In the first line, 8D102 should be:

8D0102

In line 0910, 8D0AD0 should be:

8D1AD0

Lines 0940 to 0970 should read as follows:

```
0666   8D18D0   0940   STA   COLPF2
0669   E6CF     0950   INC   DECK    NEXT DECK
066B   68       0960   PLA           RESTORE ACCUMULATOR
066C   40       0970   RTI           DONE
```

Page 75. These commands, definitions, and page references should be added to the beginning of the **QUICK REFERENCE** list of **EDITOR** commands:

DOS	switches to DOS Menu, destroying current assembler RAM program in process, unless using DOS 2.0S with MEM.SAV (2 page reference in this manual)	
SIZE	gives memory map buffer addresses for edit text buffer and user RAM (start of buffer, end of used area, end of available area)	6
LOMEM	bumps the start address edit text buffer (your source program) up or down in memory	7

The second to the last **EDITOR** command should read:

SAVE #C: <xxxx, yyyy

Page 77. Here is the corrected modification program:

```
EDIT
10        *=600
20        LDA     #$FF
30        STA     $2441
40        STA     $2448
50        STA     $14BF
60        STA     $14C0
70        .END
```

At the end of the listing, an additional line that indicates zero errors will appear.