

Atari Assembler Editor#

Copyright (C) 1980 Atari, Inc. & Kathleen Ann O'Brien

Table of Contents

- [Atari Assembler Editor](#)
- [Background](#)
- [Assembler Editor - revision A & B cartridges, outlook for revision C and maybe D](#)
- [CAR images](#)
- [ROM images](#)
- [BIN images](#)
- [ATR images](#)
- [XEX files](#)
- [Manuals](#)
- [Pictures](#)
- [References](#)
- [Changing the editors color and background](#)
- [Hello World! Source Code for the Atari Assembler Editor](#)
- [Atari Assembler Editor Mnemonics](#)
- [Commands and error codes](#)

Background#

Atari hired Shepardson Microsystems to write [Atari BASIC](#) for the 8-bit line. Shepardson developed it using a cross-compiler, but took the opportunity to begin writing their own assembler for the Atari platform as well. This was released by Atari in ROM cartridge form in 1980.

Atari Assembler Editor shared many components with Atari BASIC, notably the screen editor which used the same line-number based system as BASIC. However, it added several editing commands, including a REnumber and DELete, which, perhaps surprisingly, could also be used to edit BASIC programs.

The system ran entirely in RAM, meaning that both the source code and resulting machine code had to be able to fit in memory at the same time. This could be a significant limitation in many situations. Additionally, it did not include the ability to link multiple files together into a single larger program, which put further limits on the sort of programs that could be developed with it. [Eastern Front 1941](#), which was about 12 to 16k of machine code, required the source to be broken into six modules and then linked together by hand using DOS. Significant effort was needed to ensure that the memory locations in the six files were correct.

For larger programs, Atari also sold the [Atari Macro Assembler and Program-Text Editor CX8121](#), which used a separate full-screen editor, saved files to disk, and included a linker. This was, however, both slow and expensive. As a result, many programmers were left wanting something more powerful than Assembler Editor (notably with macro support, which it lacked) but faster and less expensive than Macro Assembler. This led to a thriving market for 3rd party assemblers on the Atari platform.

[Table of Contents](#)

Atari Assembler Editor Box Cover

The Atari Assembler Editor cartridge was a program used to edit, compile and debug assembly language programs for the Atari 8-bit computers. It was programmed by Kathleen Ann O'Brien of

Shepardson Microsystems, Inc., later founding member of OSS, Inc.. It was the first commercially available assembler for the Atari 8-bit computers ever. And yes, it was programmed by a woman. Showing once more, how far ahead of time SMI and OSS were. The program was a two-pass 6502 assembler, in an 8 KB cartridge. With the command **SIZE**, the user gets the info, how much space is free and the command **DOS**, exits the cartridge and jumps into the Disk Operating System (DOS). There are rumors of a **LOMEM** command, but AtariWiki couldn't verify this.

Assembler Editor - revision A & B cartridges, outlook for revision C and maybe D#

There are two different versions of the cartridge which could be verified very easy, please see the md5 checksums below. AtariWiki dares to define them as revision A and B in analogy to the Atari Basic cartridge, which was delivered in three different versions. It is easy to find out, which version one has, just type in ASM and hit RETURN. If you have a revision A cartridge, no error message is shown, in case you have a revision B cartridge, the amount of errors are shown. Please see the two pictures below for an example:

Assembler Editor - revision A cartridge test result

Assembler Editor - revision B cartridge test result

A binary compare of both cartridges did show, that there is much more different than that! Future investigations have to show, what else is different and how the user can profit of it.

Revision C:

In the future all bugs should be collected and if they can be fixed, a revision C is thinkable. Of course, this requires the source code of the program, which we sadly do not have, nor a license for. Maybe there will be a good soul out there, who will give it to us in the future? At least, the [source code chapter](#) has shown, that there is always hope, unless we never give up. :-)

Revision D:

If we go far ahead of this, then even an interbreed with the [TURBO-BASIC XL source code](#) is thinkable. In 1985 this was done for the Atari Basic cartridge by Frank Ostrowski and because many code routines are similar, there is a great chance to achieve this. Of course, even a renaming of the cartridge is possible, AtariWiki suggest to call it: 'Atari Turbo-Assembler Editor'. ;-)

CAR images#

- [Assembler Editor - Revision A.car](#)
- [Assembler Editor - Revision B.car](#)

ROM images#

- [Assembler Editor - Revision A.rom](#) ; md5 checksum: 13b6dfbcad3ee35f511209b790e04675
- [Assembler Editor - Revision B.rom](#) ; md5 checksum: aaf134cca657e21e2ad692b03a3233e0

BIN images#

- [Assembler Editor - Revision A.bin](#)
- [Assembler Editor - Revision B.bin](#)

ATR images#

- [Atari Assembler Editor with DOS 2.0S.atr](#) ; Assembler Editor - Revision A & B with Atari DOS 2.0S
- [Atari Assembler Editor with OSS DOS XL 2.30p Color.atr](#) ; Assembler Editor - Revision A & B with OSS DOS XL 2.30p Color

- [Hello World Source Code Atari Assembler-Editor with DOS 2.0S SD.atr](#)
- [Hello World Source Code Atari Assembler-Editor with DOS 2.75 SD.atr](#)

XEX files#

- [Assembler Editor - Revision A.xex](#) ; Assembler Editor - Revision A execution file
- [Assembler Editor - Revision B.xex](#) ; Assembler Editor - Revision B execution file

Manuals#

- [ATARI Assembler Editor User's Manual with OCR](#) ; size: 5.8 MB ; good quality
- [ATARI Assembler Editor User's Manual](#)
- [ATARI Assembler Editor User's Manual Update](#)
- [ATARI Assembler Editor User's Manual Errata 1](#) ; size: 20.6 MB ; Many thanks to Atarimania!
- [ATARI Assembler Editor User's Manual Errata 2](#) ; size: 238 KB
- [ATARI Assembler Editor Reference Card](#)
- [ATARI Assembler Editor Command Summary](#)
- [The Atari Assembler Editor Reference.rtf](#) ; The Atari Assembler Editor Reference by Matthew J. W. Ratcliff
- [The Atari Assembler Editor bug list.rtf](#) ; List of known bugs in the program, with description, solution, work around and source

- [The Atari Atari Assembler Editor-Don Inman, Kurt Inman](#) ; Mega-Thanks to Atarimania for hosting!!!

- [Der Atari Atari Assembler Editor-Don Inman, Kurt Inman](#) ; Mega-Thanks to Atarimania for hosting!!!

Pictures#

Atari Assembler Editor Box Cover - front

Atari Assembler Editor Box Cover - back

Assembler Editor Cartridge - Revision A

Assembler Editor Cartridge - Revision B

Atari Assembler Editor Box Cover - front German version

Atari Assembler Editor Box Cover - back German version

References#

- <http://ataripodcast.libsyn.com/antic-interview-21-the-atari-8-bit-podcast-kathleen-obrien-oss> ; ANTIC Interview 22 - The Atari 8-bit Podcast - with Kathleen Ann O'Brien from OSS, Inc.
- https://computingpioneers.com/index.php/Kathleen_O%27Brien ; transcript of the interview above

- <http://a8preservation.com/#/software/title/%2Fapi%2Ftitles%2F587> ; Atari Assembler Editor on a8preservation.com
- <http://www.atarimagazines.com/hi-res/v1n1/advanceduser.php> ; Atari Assembler Editor in comparison to other assemblers by Russ Wetmore
- http://open-source.ma.web.id/IT/1634-1523/Atari-Assembler-Editor_5818_open-source-ma.html ; good info page about the Atari Assembler Editor including the Hello World! source code
- [Atari Assembler Editor discussion on AtariAge](#) ; discussion on AtariAge about the Atari Assembler Editor

Changing the editors color and background#

Assembler Editor Cartridge - changing the editor's color and background to user desired colors for better editing

Hello World! Source Code for the Atari Assembler Editor#

This is the Hello World! source code:

```

10 ; HELLO.ASM
20 ; -----
30 ;
40 ; THIS ATARI ASSEMBLY PROGRAM
50 ; WILL PRINT THE "HELLO WORLD"
60 ; MESSAGE TO THE SCREEN
70 ;
0100 ; CIO EQUATES
0110 ; =====
0120     *= $0340     ;START OF IOCB
0130 IOCB
0140 ;
0150 ICHID *= *+1     ;DEVICE HANDLER
0160 ICDNO *= *+1     ;DEVICE NUMBER
0170 ICCOM *= *+1     ;I/O COMMAND
0180 ICSTA *= *+1     ;I/O STATUS
0190 ICBAL *= *+1     ;LSB BUFFER ADDR
0200 ICBAH *= *+1     ;MSB BUFFER ADDR
0210 ICPTL *= *+1     ;LSB PUT ROUTINE
0220 ICPTH *= *+1     ;MSB PUT ROUTINE
0230 ICBLL *= *+1     ;LSB BUFFER LEN
0240 ICBLH *= *+1     ;MSB BUFFER LEN
0250 ICAX1 *= *+1     ;AUX BYTE 1
0260 ICAX2 *= *+1     ;AUX BYTE 1
0270 ;
0280 GETREC = 5       ;GET TEXT RECORD
0290 PUTREC = 9       ;PUT TEXT RECORD
0300 ;
0310 CIOV = $E456     ;CIO ENTRY VECTOR
0320 RUNAD = $02E0    ;RUN ADDRESS
0330 EOL = $9B        ;END OF LINE
0340 ;
0350 ; SETUP FOR CIO
0360 ; -----
0370     *= $0600
0380 START LDX #0     ;IOCB 0

```

```

0390     LDA #PUTREC ;WANT OUTPUT
0400     STA ICCOM,X ;ISSUE CMD
0410     LDA #MSG&255 ;LOW BYTE OF MSG
0420     STA ICBAL,X ; INTO ICBAL
0430     LDA #MSG/256 ;HIGH BYTE
0440     STA ICBAH,X ; INTO ICBAH
0450     LDA #0      ;LENGTH OF MSG
0460     STA ICBLH,X ; HIGH BYTE
0470     LDA #$FF   ;255 CHAR LENGTH
0480     STA ICBLH,X ; LOW BYTE
0490 ;
0500 ; CALL CIO TO PRINT
0510 ; -----
0520     JSR CIOV   ;CALL CIO
0530     RTS       ;EXIT TO DOS
0540 ;
0550 ; OUR MESSAGE
0560 ; -----
0570 MSG .BYTE "HELLO WORLD!",EOL
0580 ;
0590 ; INIT RUN ADDRESS
0600 ; -----
0610     *=   RUNAD
0620     .WORD START
0630     .END

```

This is the Hello World! source code in assembled form:

```

10 ; HELLO.ASM
20 ; -----
30 ;
40 ; THIS ATARI ASSEMBLY PROGRAM
50 ; WILL PRINT THE "HELLO WORLD"
60 ; MESSAGE TO THE SCREEN
70 ;
0100 ; CIO EQUATES
0110 ; =====
0000 0120     *=   $0340      ;START OF IOCB
0130 IOCB
0140 ;
0340 0150 ICHID  *=   *+1      ;DEVICE HANDLER
0341 0160 ICDNO  *=   *+1      ;DEVICE NUMBER
0342 0170 ICCOM  *=   *+1      ;I/O COMMAND
0343 0180 ICSTA  *=   *+1      ;I/O STATUS
0344 0190 ICBAL  *=   *+1      ;LSB BUFFER ADDR
0345 0200 ICBAH  *=   *+1      ;MSB BUFFER ADDR
0346 0210 ICPTL  *=   *+1      ;LSB PUT ROUTINE
0347 0220 ICPH   *=   *+1      ;MSB PUT ROUTINE
0348 0230 ICBLH  *=   *+1      ;LSB BUFFER LEN
0349 0240 ICBLH  *=   *+1      ;MSB BUFFER LEN
034A 0250 ICAX1  *=   *+1      ;AUX BYTE 1
034B 0260 ICAX2  *=   *+1      ;AUX BYTE 1
0270 ;
0005 0280 GETREC =    5        ;GET TEXT RECORD
0009 0290 PUTREC =    9        ;PUT TEXT RECORD
0300 ;
E456 0310 CIOV   =   $E456     ;CIO ENTRY VECTOR
02E0 0320 RUNAD  =   $02E0     ;RUN ADDRESS
009B 0330 EOL    =   $9B       ;END OF LINE
0340 ;

```

```

0350 ; SETUP FOR CIO
0360 ; -----
034C 0370      *=   $0600
0600 A200 0380 START LDX #0      ;IOCB 0
0602 A909 0390      LDA #PUTREC  ;WANT OUTPUT
0604 9D4203 0400     STA ICCOM,X  ;ISSUE CMD
0607 A91F 0410     LDA #MSG&255  ;LOW BYTE OF MSG
0609 9D4403 0420     STA ICBAL,X  ; INTO ICBAL
060C A906 0430     LDA #MSG/256  ;HIGH BYTE
060E 9D4503 0440     STA ICBAH,X  ; INTO ICBAH
0611 A900 0450     LDA #0        ;LENGTH OF MSG
0613 9D4903 0460     STA ICBLH,X  ; HIGH BYTE
0616 A9FF 0470     LDA #$FF     ;255 CHAR LENGTH
0618 9D4803 0480     STA ICBLL,X  ; LOW BYTE
0490 ;
0500 ; CALL CIO TO PRINT
0510 ; -----
061B 2056E4 0520     JSR CIOV     ;CALL CIO
061E 60 0530      RTS          ;EXIT TO DOS
0540 ;
0550 ; OUR MESSAGE
0560 ; -----
061F 48 0570 MSG      .BYTE "HELLO WORLD!",EOL
0620 45
0621 4C
0622 4C
0623 4F
0624 20
0625 57
0626 4F
0627 52
0628 4C
0629 44
062A 21
062B 9B
0580 ;
0590 ; INIT RUN ADDRESS
0600 ; -----
062C 0610      *=   RUNAD
02E0 0006 0620      .WORD START
02E2 0630      .END
0 ERRORS

```

Atari Assembler Editor Mnemonics#

```

ADC Add Memory to Accumulator with Carry
AND AND Accumulator with Memory
ASL Shift Left (Accumulator or Memory)
BCC Branch if Carry Clear
BCS Branch if Carry Set
BEQ Branch if Result = Zero
BIT Test Memory Against Accumulator
BMI Branch if Minus Result
BNE Branch if Result ? Zero
BPL Branch on Plus Result
BRK Break
BVC Branch if V Flag Clear
BVS Branch if V Flag Set
CLC Clear Carry Flag

```

CLD Clear Decimal Mode Flag
CLI Clear Interrupt Disable flag (Enable Interrupt)
CLV Clear V Flag
CMP Compare Accumulator and Memory
CPX Compare Register X and Memory
CPY Compare Register Y and Memory
DEC Decrement Memory
DEX Decrement Register X
DEY Decrement Register Y
EOR Exclusive-OR Accumulator with Memory
INC Increment Memory
INX Increment Register X
INY Increment Register Y
JMP Jump to New Location
JSR Jump to Subroutine
LDA Load Accumulator
LDX Load Register X
LDY Load Register Y
LSR Shift Right (Accumulator or Memory)
NOP No Operation
ORA OR Accumulator with Memory
PHA Push Accumulator on Stack
PHP Push Processor Status Register (P) onto Stack
PLA Pull Accumulator from Stack
PLP Pull Processor Status Register (P) from Stack
ROL Rotate Left (Accumulator or Memory)
ROR Rotate Right (Accumulator or Memory)
RTI Return from Interrupt
RTS Return from Subroutine
SBC Subtract Memory from Accumulator with Borrow
SEC Set Carry Flag
SED Set Decimal Mode Flag
SEI Set Interrupt Disable Flag (Disable Interrupt)
STA Store Accumulator
STX Store Register X
STY Store Register Y
TAX Transfer Accumulator to Register X
TAY Transfer Accumulator to Register Y
TSX Transfer Register SP to Register X
TXA Transfer Register X to Accumulator
TXS Transfer Register X to Register SP
TYA Transfer Register Y to Accumulator

Commands and error codes#

Atari Assembler Editor - Editor commands

Atari Assembler Editor - Assembler commands

Atari Assembler Editor - Debugger commands

Atari Assembler Editor - Error Codes 001-139

Atari Assembler Editor - Error Codes 140-165

Atari Assembler Editor - Error Codes 2-171