Bug/65 Version 2.0 (C) 1982 McStuff Co. and Optimized Systems Software, Inc.[#](#)

Powerful, individual adaptable debugger for Atari 8 bit computers

If you own Rev. 1.1 of the software, not the manual, please let us know. We can make you an offer, you can't resist. ;-) The version we search for is maybe just from McStuff Co. and not from OSS. Thank you so much in advance for your help, greatly appreciated by the community.

Table of Contents

Image[#](#)

Bug/65 Version 2.0 (C) 1982 McStuff Company and Optimized Systems Software, Inc.

BUG/65 Disassembly[#](#)

- [Bug-65 Disassembly.txt](#)

CAS-Image[#](#)

- [OSS_BUG_65_2.0.cas](#)

ATR-Images[#](#)

- [OSS_Bug-65_with_OSS_DOS_XL_2.30p_2000-9A00.atr](#) ; all versions, normal and relocatable, further with version 4 patch and the User Command Handler Example, please see below
- [OSS_Bug-65_with_OSS_DOS_XL_2.30p_2000-9000_and_3000_Color.atr](#) ; same as above, but with the 3000 color version instead of the 9A00 version
- [MAC-65_2.00_and_4.20_with_Bug-65_2.0_normal_and_V4_ready_and_DOS_XL_2.30.atr](#) ; normal and relocatable, further with already patched version 4 of BUG/65 and the User Command Handler Example, please see below

Manuals[#]

- [BUG/65 Manual Rev. 1.0](#) ; size: 2.4 MB ; original first manual from OSS of BUG/65 ; thank you so much DjayBee from AtariAge, that is a great find! Further on [archive.org](#)
- [BUG/65 Manual Rev. 1.0](#) ; size: 1.3 MB ; original first manual from OSS of BUG/65 with table of contents, but without section 13: error messages, appendix and the last 13 pages in whole
- [BUG/65 Manual Rev. 1.1](#) ; size: 987 KB ; same as above but with: table of contents, section 13: error messages and appendix
- [BUG/65 Manual Rev. 1.1 with ERRATA](#) ; size: 2.7 MB ; same as above but with an errata page: 'ERRORS IN YOUR BUG/65 MANUAL' (page 4 in the pdf file) for the manual itself ; as of this moment, it is not understood, why the there mentioned commands: N, O, R', R", W' and W" do not work with the above version 2.0 of BUG/65. Either we have version 1.1 for real, while 2.0 is shown at start or something else is wrong here. Anyway, the mentioned pages on the error page do match with pages in the rest of the manual. Further, the below mentioned BUG/65 version 4 patch should not be done, according to the error page from December 1983(?), while it should be done according to the OSS newsletter from summer 1983. Maybe just the 'old boys club' from the golden age can solve this? Any help in that case is very welcome at any time. We would really appreciate if you may can help us here. :-)
- [BUG/65 Manual Rev. 1.1 with ERRATA](#) ; size: 23.6 MB ; complete re-edit of the above with navigation bar, hyperlinks and OCR maintaining the original form

Please take into account, that in the manuals above Rev. 1.1 is mentioned, while just version 2.0 of the software is available up to now.

- [BUG-65 Manual in html](#) ; BUG/65 manual from above in html with css ; Mega-thanks to Erhard Pütz and Dieter Lichtenegger for giving us the best BUG/65 manual possible in html format! :-)
- [BUG-65 Manual 1.pdf](#) ; size: 102 KB ; with yellow frame taken from the software ; BUG/65 manual from above in pdf form with navigation bar and hyperlinks ; Mega-thanks to Erhard Pütz and Dieter Lichtenegger for giving us the best BUG/65 manual possible in pdf format! :-)
- [BUG-65 Manual 2.pdf](#) ; size: 283 KB ; without yellow frame, just black & white, but with some colors e. g. chapters
- [BUG-65 Manual 3.pdf](#) ; size: 213 KB ; with yellow frame taken from the software, but without navigation bar and hyperlinks
- [BUG-65 Manual 4.pdf](#) ; size: 233 KB ; black & white version without navigation bar and hyperlinks

BUG/65 User Command Handler Example[#]

```
0005 ;***********************************************
0010 ;   EQUATES INTO BUG/65:
0015 ;
0020 loadpoint = ????            ; to be determined by user!!
0025 lp        = loadpoint       ; just an abbreviation
0030 MCBEND    = lp+$21F         ; BUG/65 END CODE MSB
0035 DISPV     = lp+$209         ; DISPLAY CHAR
0040 USRCMD    = lp+$220         ; USER COMMAND VECTOR
0045 GET2HX    = lp+$22C         ; GET 2 HEX PARAMS
0050 HEX1      = $FC             ; HEX PARAM 1 RESULT
0055 HEX2      = $FE             ; HEX PARAM 2 RESULT
0060 ERRPAR    = lp+$235         ; REPORT PARAM ERROR
0065 DHXBYT    = lp+$238         ; DISPLAY HEX BYTE
0070 LSTPG0    = lp+$240         ; LAST BUG/65 P0 BYTE USED
0075 EOL       = $9B             ; END OF LINE CHAR
0080 ;
0090 ;***********************************************
0100      *=  USRCMD            ; PATCH US INTO BUG/65
```

```
0110      JMP USERC1
0120 ;
0130      *=  lp+$2000     ; RIGHT AFTER BUG/65 CODE
0140 USERC1 CMP #'1        ; COMMAND "1" ?
0150      BEQ CMDOK        ; YES
0160      RTS              ; ELSE RTN EQUAL RESET - ERR
0170 ;
0180 CMDOK JSR GET2HX      ; GET START, END
0190      LDA HEX1         ; MAKE SURE BOTH SPECIFIED
0200      ORA HEX1+1
0210      BEQ PARMER       ; OR ELSE ERROR
0220      LDA HEX2
0230      ORA HEX2+1
0240      BNE PARMOK
0250 ;
0260 PARMER JMP ERRPAR     ; REPORT PARAM ERROR
0270 ;
0280 PARMOK LDX LSTPG0     ; LAST BUG/65 P0 BYTE
0290 ;                          (WE'LL USE THE NEXT
0300 ;                           FOR OUR ACCUMULATOR)
0310      LDA #0           ; CLEAR ACCUMULATOR
0320      STA 1,X
0330      TAY              ; INIT Y PTR INDEX
0340 ;
0350 LOOP LDA HEX2+1       ; PAST END ADDRESS ?
0360      CMP HEX1+1
0370      BCC DONE         ; YES
0380      BNE NXTEOR       ; NO
0390      LDA HEX2
0400      CMP HEX1
0410      BCC DONE         ; YES
0420 ;
0430 NXTEOR LDA (HEX1),Y   ; CALC EOR CHKSUM
0440      EOR 1,X          ; EOR WITH ACCUM
0450      STA 1,X          ; AND SAVE IN ACCUM
0460      INC HEX1         ; BUMP PTR
0470      BNE LOOP
0480      INC HEX1+1
0490      JMP LOOP
0500 ;
0510 DONE LDA #EOL         ; TO NEXT SCREEN LINE
0520      JSR DISPV
0530      LDX LSTPG0       ; RESTORE ACCUM ADDRESS
0540      LDA 1,X          ; DISPLAY HEX RESULT
0550      JSR DHXBYT
0560      LDA #0           ; RTN OK (EQUAL SET)
0570      RTS
0580 ;
0590      *=  MCBEND       ; CHANGE BUG/65 CODE
0600      .BYTE  >[*+$FF]  ; END BYTE TO INCLUDE
0610      .END             ; THAT'S ALL FOLKS
```

The MAC/65 file of the 'User Command Handler Example': 'USRCOMHD.M65' is already on all of the atr images above.

BUG/65 does not print with XL/XE OS[#](#)

Mega-thanks to Erhard Pütz for the final solution of this problem! :-)

Device ID $3F instead $40. BUG/65 jumps indirect into the put-byte-routine of the OS ($E436 -> $FECA+1), therefore no IOCB-channel is opened and the device number is missing.
Patch (thank you ep for the solution!):
Load BUG/65 and change the following:

```
ORG+$068B LDX #36
          LDY #E4
          JMP 2777
```

=>

```
ORG+$068B JMP 0600
0600      PHA
      LDX #10
      LDA #1
      STA 341,X
      PLA
      JMP FECB
```

BUG/65 version 4 patch[#]

The version 4 patch is available in two different kinds: a) a short BASIC program in form of a BAS-file. and b) a short machine language program in form of a COM-file.

a): BASIC program:

```
5 REM BUG/65 BUG -- VERSION 4 PATCH PROGRAM
10 XIO 36,#1,0,0,"D:BUG65.COM":REM UNPROTECT FILE
20 OPEN #1,12,0,"D:BUG65.COM"
30 FOR I=1 TO 2668:REM MOVE TO PROPER POSITION
40 GET #1,C
50 NEXT I
60 PUT #1,12
70 CLOSE #1
```

The above program named as "BUGV4FIX.BAS" is in form of a file already on all of the atr-images above.

To make the patch to BUG/65 do as follows: 1) Insert your BASIC cartridge or use BASIC A+ disk. 2) Place your BUG/65 disk into the disk drive. 3) Use this BASIC program above to apply the one byte patch so that BUG/65 will work with OS/A+ version 4.

b): machine language program:

name of file: BUGV4FIX.COM

BUGV4FIX.COM ; we are still searching for that file, it seems to be lost in time, like tears in rain. Any help, any hint in that case is very much appreciated. :-)

At the end of the BUG/65 manual we can read:
1. Copy the files BUG65.COM and BUGV4FIX.COM to a version 4 disk using the COPY24 command (see the DOSXL manual for details on this command).
2. At the version 4 "D1:" prompt, type the command: BUGV4FIX (RETURN).
3. The file BUG65.COM on that disk is now compatible with version 4 of DOSXL.
WARNING: Do NOT perform the BUGV4FIX command on your version 2 master disk!

BUG/65 patch from OSS Newsletter October 1984[#]

When BUG gets control as the result of a BRK instruction being executed, it actually checks to see it the BRK instruction is one which it placed (as a result of a "G" command). If not, it assumes that it is a user breakpoint, to be handled by a user routine! Guess what? Most of us don't go around adding user BRK handlers to Atari?s OS, so the machine goes off into never-never-land.

Quick hix (again, not hor the faint of heart): After loading BUG/65 from CP, perform the following actions:

1. Display the contents of location $2E7 (LOMEM).
2. Use the "H" command to add $17EE to those contents. (Example: if $2E7 contains 2200, use "H 2200 17EE? to find the sum.) Call this value ADDSUM.
3. Display the memory at ADDSUM (via the "D" command. You should see the following bytes (if not, stop now): 39EE E9 02 4D . . . ("39EE" is arbitrary-value of ADDSUM if LOMEM is at $2200.)
4. Use BUG/65 commands as follows:

```
Z 600
LDA #B0
STA addsum (use actual address instead of name!)
LDA #1E
STA addsum+1 (again, use an actual address)
BRK (hit the BREAK key to exit miniassembler)
G 600
```

If you did everything correctly, BUG/65 should allow a "USER RUN" and then print a breakpoint message for the BRK at $60A. If it didn?t work, you may have an already-fixed version of BUG/65, so ignore all this.

Creating a non-relocatable version[#]

In order to allow itself to be relocated virtually anywhere in memory, BUG/65 as shipped includes a relocation bit map and relocation program. In addition, relocatable BUG/65 always loads in at locations $9800 through $BC00. If these addresses are "poison" to you (e.g., if you want to use BUG/65 with a cartridge plugged in), you may wish to produce a non-relocatable version designed to run within an address range you pick. If so, USING A 48K SYSTEM, simply specify the loadpoint, as shown in the preceding section (e.g., via BUG65 7000) and allow BUG/65 to load and relocate. Then exit to OS/A+ (via Quit) and use OS/A+ intrinsic command SAVE to save a non-relocatable version. The address range to be SAVEd may be calculated as follows:

SAVE filename.com loadpoint+$200 loadpoint+$1FFF

Thus, if you had specified "BUG65 7000", you could save the non-relocatable version via

SAVE BUG7000.COM 7200 8FFF

thus also giving it a name which will later remind you where it will load at. To execute this non-relocatable version, simply type in its name (BUG7000 in the example shown).

Summary of major features of BUG/65[#]

? A full set of debugging commands - change memory, display memory, goto user program with break points, etc.
? Binary file read and write, including appended write
? A disassembler
? An instant assembler providing labelling capability
? Expanded command addressing capability: hex or decimal addresses, + and - operators supported, relocated addresses supported
? Read or write disk sector(s)
? Multiple commands permitted in a command line. Command lines can be repeated with a single keystroke or repeated forever with the special slash operator.

? Support for relocatable assemblers - the base of a module can be specified and then used to reference addresses in that module

? BUG/65 commands can be executed from a command file, and there is a command to create command files

? Hex to decimal and decimal to hex conversions provided

? Memory protection of BUG/65's code and data. BUG/65 won't allow you to use a BUG/65 command that will destroy any part of BUG/65 itself. For example, you can't use the Fill command to overwrite BUG/65's code.

? Page zero sharing. BUG/65 shares page zero with a user program by keeping two copies of the shared page zero locations - one for the user and one for BUG/65 itself.

Command Summary of BUG/65[#](#)

Command Summary of BUG/65 Version 2.0 (C) 1982 McStuff Company and Optimized Systems Software, Inc.