

MacsBug 6.2

Online Help Listing

Table of Contents

Editing.....	1
Selecting procedure names.....	1
Expressions.....	1
Values.....	2
Operators.....	2
Flow control.....	2
Breakpoints.....	3
A-Traps.....	3
Disassembly.....	4
Heaps.....	4
Symbols.....	4
Stack.....	5
Memory.....	5
Registers.....	5
Macros.....	6
Miscellaneous.....	6

Editing

Type a command and then press Return or Enter to execute it. Pressing Return without entering a command repeats the last command. You can enter more than one command on the command line by separating commands with a semicolon (;).

Editing commands

Command-V	Scroll command buffer down and copy command line to current command line.
Command-B	Scroll command buffer up and copy command line to current command line.
Option-Left Arrow	Move cursor left one word.
Option-Right Arrow	Move cursor right one word.
Option-Delete	Delete the word to the left of the cursor.
Command-Left Arrow	Move cursor to beginning of command line.
Command-Right Arrow	Move cursor to end of command line.
Command-Delete	Delete the line to the left of the cursor.

Selecting procedure names

Display a list of procedure names by typing Command-: or Command-D. Select a procedure from the list by using the up and down arrows. Copy the selected procedure name to the insertion point of the command line by pressing Return. Press Escape to leave command line unchanged. Display a selected (qualified) list of procedure names by typing the first letter(s) of the procedure name (after displaying the list with Command-: or Command-D). Pressing Delete undoes the qualification one letter at a time. Entering Command-: or Command-D undoes the qualification and shows all procedure names. With C++ and Object Pascal names you may use the Tab key to qualify the class.

Expressions

The general form of an expression is: value1 [operator value2]. Use parentheses to control the order of evaluation. Expressions always evaluate to a 32 bit value unless .W or .B follows the value. Expressions evaluate to either a numeric or a boolean value based on the operators used. The action of some commands change based on this result. For instance, BR addr expr will break each n times if expression is numeric or it will break when expr is true if expression is boolean.

Values

Registers	All 68000 family registers use their Motorola names. MMU 64 bit registers and floating point registers are not allowed in expressions.
Numbers	Numbers are hex by default but can be preceded by a '\$' in the case of conflicts with registers An and Dn. Numbers are decimal if they are preceded by a '#'. Symbols are found by searching the heap and evaluate to an address.
Symbols	
Traps	Trap number in the range A000 to ABFF or a trap name. Trap names can be preceded by a '+' in the case of conflicts with symbol names.
'.'	The last address referenced by certain commands. For instance SM sets dot to the first address that was changed and sets the last command to DM. Typing return will display the memory that was changed.
':'	The address of the start of the proc shown in the PC window. Not valid if no proc name exists for PC.

Operators

Arithmetic	+ - * / MOD
Boolean	AND or & OR or NOT or ! XOR
Equality	= or == <> or != < > <= >=
Indirection	@ (prefix) or ^ (postfix)

Flow control

G [addr]	Resume execution at addr or PC if no addr. Command-G is the same as entering G and pressing Return.
GT addr [';cmds']	Go till addr is reached and optionally execute one or more commands. The addr can be in ROM but execution will be much slower.
S [n expr]	Step n instructions or until expr is true. Command-S is the same as entering S and pressing Return.
SO [n expr]	Step n instructions or until expr is true. JSRs, BSRs and Traps are treated as one instruction. For historical reasons, T (for Trace) is allowed as an alias for SO. Command-T is the same as entering T and pressing Return. If n is zero then clear all step points.
SS addr [addr]	Step until checksum of addr range changes. If you do not specify a range, step until long word at addr changes.
MR [offset addr]	Break after the current procedure returns by replacing its return address. If the parameter is less than A6 then the return address is at A7+offset. If the parameter is greater than or equal to A6 then the return address is at addr+4. If no parameter then the

return address is at A7.

Breakpoints

BR addr [n | expr] [';cmds']

Break at addr each n times or when expr is true and optionally execute one or more commands. If no n or expr then break always. The addr can be in ROM but execution will be much slower.

BRM string

Set breakpoints at all procedure names that contain string. String can occur anywhere in the name, not just at the beginning.

BRC [addr]

Clear breakpoint at addr or all breakpoints if no addr.

BRD

Display breakpoint table.

A-Traps

Appending A to an A-Trap command name specifies that the command applies only to A-Traps that are called from the application heap. Entering two traps as parameters defines a trap range. Entering no traps defines the range A000 to ABFF.

ATB[A] [trap [trap]] [n | expr] [';cmds']

Break at traps each n times or when expr is true and optionally execute one or more commands. If no n or expr then break always.

ATT[A] [trap [trap]] [n | expr]

Display trap information each time a specified trap is called.

ATHC[A] [trap [trap]] [n | expr]

Check the heap each time a specified trap is called.

ATSS[A] [trap [trap]] [n | expr], addr [addr]

Checksum addr range each time a specified trap is called. If the checksum has changed then break into MacsBug. If no second addr then checksum the long word at addr.

ATC [trap [trap]]

Clear trap range or all traps if no parameters.

ATD

Display the A-Trap tables.

ATR[A] [ON | OFF]

Turns trap recording on or off. Toggle if no parameters. Information about the most recent trap calls is recorded.

ATP

Plays back the information recorded while ATR is on.

DSC[A] [ON | OFF | str]

Turns Discipline on or off. Toggle if no parameter. Any parameters other than ON or OFF are passed to Discipline for interpretation. Discipline examines parameters before trap calls and examines results after trap calls. Any errors break into MacsBug.

DSCX [OFF | ON]

Turn discipline breaks on or off. Default ON.

Disassembly

All commands assume the PC if no addr is specified.

IL [addr [n]]
 Disassemble n lines from addr. If no n then display half page.

IP [addr]
 Disassemble half page centered around addr.

ID [addr]
 Disassemble 1 line starting at addr.

IR [addr]
 Disassemble till the end of the routine addr is in.

DH expr ...
 Disassemble one or more exprs as a sequence of 16 bit opcodes.

Heaps

HX [addr]
 Set the current heap to the heap at addr. If no parameter then toggle between the Application, System and user heaps.

HZ [addr]
 List all known heap zones. If parameter then list all heap zones embedded into the heap at addr.

HD [F | N | R | L | P | Q | RS | TYPE]
 Display specific blocks in the current heap or all blocks if no parameter. The possible qualifiers are

- F: Free blocks
- N: Nonrelocatable blocks
- R: Relocatable blocks
- L: Locked blocks
- P: Purgeable blocks
- Q: Questionable blocks
- RS: Resource blocks
- TYPE: Resource blocks of this type

HT
 Display a summary of the current heap.

HC
 Check the current heap for inconsistencies.

HS [addr]
 Turn on scrambling of the heap at addr or ApplZone if no addr. Calling NewPtr, NewHandle, ReallocHandle, SetPtrSize or SetHandleSize checks the heap before the call. If good then the heap is scrambled. If bad then a MacsBug break is forced. Scrambling continues until next HS or a bad heap is detected.

Symbols

RN [expr]
 Set the resource file ref num qualifier to expr. If no expr then set it to curMap. Once set, all subsequent symbol references must be from a heap block with a matching file ref num. If expr is 0 then all symbols match.

SD

8

Command-: or Command-D is now used to select symbol names.

SX [ON | OFF]

Turn symbols in disassembly on or off. Toggle if no parameter.

Stack

SC6 [addr [nbytes]]

Show the calling chain based on A6 links. If no addr then the chain starts with A6. If addr then the chain starts at addr. If no nbytes then the stack base is CurStackBase. If nbytes then the stack base is addr+nbytes.

SC7 [addr [nbytes]]

Show possible return addresses on the stack. A return address is an even address that points after a JSR, BSR or A-Trap. If no addr then A7 is the stack pointer. If addr then addr is the stack pointer. If no nbytes then the stack base is CurStackBase. If nbytes then the stack base is addr+nbytes.

Memory

All commands assume the dot address if no addr is specified.

DM [addr [n | template | basic type]]

Display memory from addr for n bytes or as defined by a template or a basic type. The basic types are Byte, Word, Long, SignedByte, SignedWord, SignedLong, UnsignedByte, UnsignedWord, UnsignedLong, PString and CString.

TMP [name]

List templates names that match name. If no name then list all template names.

DP [addr]

Display memory from addr for 128 bytes.

DB [addr]

Display the byte at addr.

DW [addr]

Display the word at addr.

DL [addr]

Display the long at addr.

SM addr expr | 'string' ...

Assign values to memory starting at addr. Each value determines the assignment size. Specific sizes can be set using SB, SW or SL.

SB addr expr | 'string' ...

Assign values to bytes starting at addr.

SW addr expr | 'string' ...

Assign values to words starting at addr.

SL addr expr | 'string' ...

Assign values to longs starting at addr.

Registers

Values can be assigned to registers with commands of the form:

RegisterName := expression or RegisterName = expression

TD

Display CPU registers.

TF

Display 68881 floating point registers.

TM

Display 68851 MMU registers.

RAD

Toggle between specifying registers as RAn or RDn and specifying registers as An or Dn. The default setting does not require the R.

Macros

MC name 'expr' | expr
 Define a macro called name that expands to 'expr' or to the current value of expr.

MCC [name]
 Clear named macro or all macros if no name.

MCD [name]
 List macros that match name. If no name then list all macros.

Miscellaneous

Escape or tilde
 Toggle between the user screen and the MacsBug screen.

RB
 Unmount the boot volume and reboot.

RS
 Unmount all volumes except server volumes and reboot.

ES
 Exit the current application.

EA
 Restart the current application.

WH [addr | trap#]
 Find the name and addr of the parameter. If no parameter then assume WH PC.

HOW
 Display the reason MacsBug was entered this time.

F addr nbytes expr | 'string'
 Search from addr to addr+nbytes-1 for the pattern. If pattern is an expr then the width of the pattern is the smallest unit (byte, word or long) that contains its value.

FB addr nbytes expr
 Search from addr to addr+nbytes-1 for the byte.

FW addr nbytes expr
 Search from addr to addr+nbytes-1 for the word.

FL addr nbytes expr
 Search from addr to addr+nbytes-1 for the long.

FP addr nbytes expr
 Search from addr to addr+nbytes-1 for the pointer.

CS [addr [addr]]
 Checksum addr range and store the value. CS without parameters checksums the last addr range and compares it to the last value. If no second addr then checksum the long word at addr.

LOG [pathname | Printer]
 Log all MacsBug output to a file or to an ImageWriter printer. LOG without parameters turns logging off.

SHOW [addr | 'addr' [L | LA | W | A]]
 Display memory at addr in status region. Quoting addr causes it to be evaluated each time the display is updated. SHOW without parameters cycles thru the display formats. The formats are
 L: Long words
 LA: Combination of Longs and ASCII

W: Words

A: ASCII Text

Miscellaneous (cont.)

SWAP

If you have a single screen then SWAP toggles between:

 Drawing step and A-Trap trace info without swapping.

 Draw step and A-Trap trace info and swap each time.

If you have multiple screens then SWAP toggles between:

 MacsBug screen is always visible.

 MacsBug screen is visible only at breaks.

DX [ON | OFF]

Turn user breaks (A9FF, ABFF) on or off. Toggle if no parameter.

DV

Display MacsBug version.

HELP [cmd | topic]

Display info about a specific command or topic. If no parameter then display all topics.