AIM 65 TEXT EDITOR COMMANDS

ENTER AND EXIT EDITOR COMMANDS

E — Enter and Initialize Editor
<E>
EDITOR
FROM = [ADDRESS] TO = [ADDRESS]
IN = [INPUT DEVICE]
Note: Defaults are TO = $0200,
FROM = Last contiguous RAM, IN = Keyboard

Q — Exit the Text Editor and Return to Monitor
= <Q>

LINE ORIENTED COMMANDS

R — Read Lines into Text Buffer from Input Device
= <R>
IN = [INPUT DEVICE]

I — Insert One Line of Text Ahead of Active Line
<I>
INSERTED TEXT LINE
ACTIVE LINE OF TEXT

K — Delete Current Line of Text
= <K>
DELETED LINE OF TEXT
ACTIVE LINE OF TEXT

U — Move the Text Pointer Up One Line
= <U>
PRIOR LINE OF TEXT

D — Move the Text Pointer Down One Line
= <D>
NEXT LINE OF TEXT

T — Move the Text Pointer to the Top of the Text
= <T>
TOP LINE OF TEXT

B — Move the Text Pointer to the Bottom of the Text
= <B>
BOTTOM LINE OF TEXT

L — List Lines of Text to Output Device
= <L>
/ [DECIMAL NUMBER]

SPACE — Display the Active Line
= < >
ACTIVE LINE OF TEXT

STRING ORIENTED COMMANDS

F — Find a Character String
= <F>
[CHARACTER STRING]
LINE CONTAINING CHARACTER STRING

C — Change a Character String
= <C>
[OLD STRING]
LINE CONTAINING OLD STRING
TO = [NEW STRING]
SAME LINE, WITH NEW STRING

ASSEMBLER ERROR CODES

01	Undefined Symbol or Label
02	Label Previously Used or Forward Reference to Page 0 Symbol
03	Illegal or Missing Opcode
04	Invalid Address
05	Accumulator Mode Invalid
06	Forward Reference to Page Zero Symbol
07	Invalid Source Statement
08	Label Begins With Numeric Character
09	Label Longer Than Six Characters
10	Non-Alphanumeric Label or Opcode
11	Forward Reference in Equate
12	Invalid Index — Must Be X or Y
13	Invalid Expression
14	Undefined Assembler Directive
15	Invalid Page 0 Operand
17	Relative Branch Out of Range
18	Illegal Operand Type for This Instruction
19	Invalid Indirect Pointer
20	A, X, Y, S and P are Reserved Names
21	Location Counter Negative — Reset P to 0

AIM 65 ASSEMBLER

ASSEMBLER COMMAND SUMMARY

<N>
ASSEMBLER
FROM = [ADDRESS] TO = [ADDRESS]
IN = [INPUT DEVICE]
LIST? [Y, N]
LIST-OUT = [OUTPUT DEVICE]
OBJ? [Y, N] **Note:** N = Object code to Memory
OBJ-OUT = [OUTPUT DEVICE] **Note:** Prompts only on Y response to OBJ?
PASS 1
SYM TBL OVERFLOW
ASSEMBLER } Displayed only if Symbol Table overflows
PASS 2
= = AAAA LABEL } Displayed only if
OBJECT CODE MNEMONIC OPCODE } LIST?Y, or LIST?N
SYMBOLIC OPERAND ;COMMENT } and error detected
* * ERROR NN **Note:** Error code displayed only on error
ERRORS = MMMM Decimal count of errors detected

ASSEMBLER EXPRESSIONS

ELEMENTS

Numeric constants — may be written in one of four bases.

Prefix Character	Base
(none)	10 (Decimal)
$	16 (Hexadecimal)
@	8 (Octal)
%	2 (Binary)

OPERATORS

Type	Operator	Operation
Arithmetic	+	Addition
Arithmetic	–	Subtraction
Special	>	High-Byte Selection
Special	<	Low-Byte Selection

Operators < and > truncate a two-byte value to its low or high byte, respectively.

ASSEMBLER DIRECTIVES

= — Assigns the value of an operand containing no forward references to either a symbol or the location counter.

{ SYMBOL } = Operand
{ * }

.BYTE — Assigns multiple ASCII strings or expressions to consecutive single byte memory locations.

.BYT Expression, Expression, . . . Expression

.WORD — Assigns multiple expression operands to consecutive memory locations in low-byte, high-byte order.

.WOR Expression, Expression, . . . Expression

.DBYTE — Assigns multiple expression operands to consecutive double byte (16 bits) memory locations in high-byte, low-byte order.

.DBY Expression, Expression, . . . Expression

.PAGE — Generates a title under a dashed line.

.PAG { 'NEW TITLE' } (New Title)
{ BLANK } (No Change of Title)
{ ' ' } (Blanks Title)

.SKIP — Generates one blank line.

.SKI

.OPT — Controls assembly listings. All are optional and can be specified in any order or in separate statements.

.OPT { LIS } , { GEN } , { ERR }
{ NOL } { NOG } { NOE }

.FILE — Last record in a multiple file source program (except the last file) which points to the continuation file.

.FIL File Name

.END — Last record in a single or multiple source file.

.END