



**MOTOROLA**

M68EDITORM(D3)

**M6800/M6809  
EDITORM  
RESIDENT EDITOR  
REFERENCE MANUAL**

**SYSTEMS**

**MICROSYSTEMS**

M6800/6809

EDITORM

RESIDENT EDITOR

REFERENCE MANUAL

The information in this document has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Furthermore, Motorola reserves the right to make changes to any products herein to improve reliability, function, or design. Motorola does not assume any liability arising out of the application or use of any product or circuit described herein; neither does it convey any license under its patent rights or the rights of others.

EXORciser, EXORtem, EXORdisk, and MDOS are trademarks of Motorola Inc.



## TABLE OF CONTENTS

Page

CHAPTER 1	GENERAL INFORMATION	
1.1	INTRODUCTION	1-1
1.2	MINIMUM CONFIGURATION FOR EDITOR OPERATION	1-1
1.3	TERMINAL PREPARATION	1-1
1.4	LOADING THE DISK OPERATING SYSTEM (MDOS)	1-2
1.5	INVOKING THE EDITOR	1-2
1.6	MODES OF OPERATION	1-3
1.6.1	CRT-MODE	1-4
1.6.2	SCROLL-MODE	1-9
CHAPTER 2	EDIT COMMANDS	
2.1	INTRODUCTION	2-1
2.2	COMMAND SYNTAX	2-1
2.3	1-9999 COMMAND	2-2
2.4	CHANGE COMMAND	2-3
2.5	DELETE COMMAND	2-6
2.6	DUPLICATE COMMAND	2-7
2.7	E COMMAND	2-8
2.8	EXTEND COMMAND	2-9
2.9	FIND COMMAND	2-10
2.10	INSERT COMMAND	2-12
2.11	LIST COMMAND	2-13
2.12	MERGE COMMAND	2-14
2.13	MOVE COMMAND	2-15
2.14	NUMBER COMMAND	2-16
2.15	PRINT COMMAND	2-17
2.16	QUIT COMMAND	2-18
2.17	RANGE COMMAND	2-19
2.18	RESEQUENCE COMMAND	2-20
2.19	SAVE COMMAND	2-21
2.20	SEARCH COMMAND	2-22
2.21	TAB COMMAND	2-23
2.22	VERIFY COMMAND	2-24
2.23	XTRACT COMMAND	2-25
2.24	BLOCK/LINE MERGE RULES	2-26
2.25	EDITOR MESSAGES	2-27
APPENDIX A	EDITING IN THE CRT MODE AND ASSEMBLY OF PROGRAMS	A-1
APPENDIX B	EDITING IN THE SCROLL MODE AND ASSEMBLY OF PROGRAMS	B-1

### LIST OF ILLUSTRATIONS

FIGURE 1-1.	Entering CRT Editor from MDOS	1-4
1-2.	CRT-MODE (Command Editing)	1-5
1-3.	CRT-MODE (Page Editing)	1-9
1-4.	SCROLL MODE	1-10

### LIST OF TABLES

TABLE 1-1.	Edit Commands	1-6
1-2.	Command Editing (CRT-MODE)	1-7
1-3.	Page Editing (CRT-MODE)	1-8
1-4.	Command Editing (SCROLL-MODE)	1-10
2-1.	Operator Messages	2-27



## CHAPTER 1

### GENERAL INFORMATION

#### 1.1 INTRODUCTION

Motorola's M6800/M6809 EDITORM Resident Editor provides users of Motorola development systems with powerful text editing capabilities. The Editor simplifies both the initial text and source program entry, as well as subsequent program modification.

The Editor permits many modes of operation. As a disk-based editor, text may be entered either from a diskette file or terminal keyboard. The Editor will operate with any TTY-compatible terminal. Use of the Editor in conjunction with the EXORterm 220 or EXORtem 155/EXORciser system allows the user to perform editing, employing specifically designed features of the EXORterm.

The Editor provides a standard set of editing functions, as found in most microprocessor editors:

- . insert string/line
- . change string
- . list program
- . delete string/line

In addition, the Editor allows more powerful editing functions, such as:

- . multi-string replacement
- . block moves
- . verification of editing changes
- . CRT-oriented text modification
- . editing ranges
- . tabbing

#### 1.2 MINIMUM CONFIGURATION FOR EDITOR OPERATION

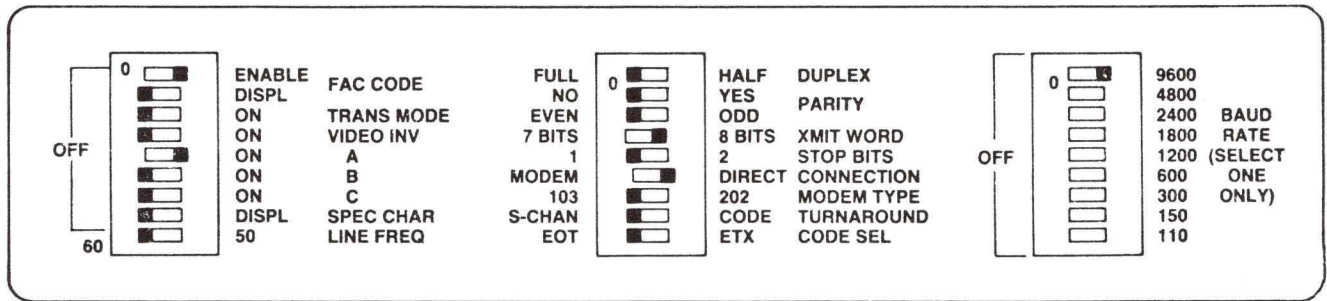
As a minimum configuration, the Editor requires the following:

- . TTY-compatible terminal or EXORterm 220 or EXORterm 155/EXORciser system
- . 32K Random Access Memory
- . EXORDisk II or EXORDisk III
- . MDOS 3.xx operating system

The use of additional memory or a line printer is optional.

#### 1.3 TERMINAL PREPARATION

The heart of the EXORterm is the CRT terminal, which houses the actual microcomputer, memory, interfaces, and other elements. Considerable flexibility is designed into the CRT to accommodate a variety of different environments and configurations. Three groups of option switches on the back of the terminal housing itself must be set as follows for proper operation of the Editor.



#### 1.4 LOADING THE DISK OPERATING SYSTEM (MDOS)

The actual procedure for initially loading the MDOS Disk Operating System will vary slightly between different hardware configurations. Refer to the EXORDISK II/III Operating System User's Guide for the exact initializing procedure for your hardware configuration. Typical procedures are:

```

EXORterm 220      depress restart key
                  EXBUG 2.x
(or equivalent   *E MDOS (CR)
TTY compatible   MDOS 3.xx
terminal)        =

```

The equal sign is an MDOS prompt character. MDOS commands may now be entered.

#### 1.5 INVOKING THE EDITOR

The MDOS command E will invoke the Editor to create or modify a named file. Following is the syntax for the edit command. All elements bounded by brackets are optional; e.g., [;options]. All elements outside the brackets must be entered. All user input is terminated by a carriage return (RETURN).

Syntax: E FILENAME1[, [FILENAME2] [, :DRIVE]] [; [TS] [FN] [S]]

where: FILENAME1 is the file to be edited, if it exists. If it does not exist, it will be created and made available for editing.

FILENAME2 is the file name of the edited result. FILENAME2 is valid only if FILENAME1 exists. If FILENAME2 is not specified, the original FILENAME1 will be deleted at the end of the edit process, and the edited result will be renamed FILENAME1. If FILENAME2 is specified, FILENAME1 is retained as an MDOS file without any alteration by the Editor. FILENAME1 uses the standard MDOS default parameters of .SA for suffix and :0 for drive number. Any portion of FILENAME2 not specified will default to the corresponding portion of FILENAME1 after it is processed.

DRIVE specifies the drive on which to put the second scratch file. The Editor will pick the optimum drive if DRIVE is not specified. The Editor uses temporary files named SCRATCH1 and SCRATCH2. SCRATCH1 is used whenever FILENAME1 exists. When the edit is terminated via QUIT or a SAVE command, SCRATCH1 is renamed FILENAME2 (or FILENAME1 if FILENAME2 was not specified). SCRATCH2 is used to support multiple edit passes in a single edit session, and is deleted after the edit is terminated by either a QUIT or E command.

TS specifies the initial tab stop settings for the Editor as follows:

- A implies assembler tabs (8,15,24) with the space bar as a special tab character and whereby an "\*" in column 1 will disable the special tab character.
- B implies FORTRAN tabs (7) with the space bar as a special tab character and whereby a "C" in column 1 will disable the special tab character.
- C implies COBOL tabs (6,9,12) with the space bar as a special tab character.

FN is the file numbering option. An N implies edit with line numbers. -N implies edit without line numbers. FN defaults to N when FILENAME1 does not exist, and is determined by the first line in FILENAME1 when it does exist. Whenever a line-numbered file is edited without using line numbers, the line numbers are treated as data.

S specifies that the Editor is to be initialized in the scroll mode, and that the user is editing from a non-EXORterm type device. With this option, the user is restricted to scroll mode command editing.

The response to the E command is shown in the following example:

```
=E LTR2 (CR)
MDOS EDITOR RELEASE xx.xx      (xx.xx is the version number)
COPYRIGHT BY MOTOROLA 1978
```

When FILENAME1 exists, the Editor will check to determine if there is sufficient disk space to hold the edited result and the work file on the drives specified. If there is not sufficient space to store an edited file the size of which is identical to the input file, the Editor will display "INSUFFICIENT DISK SPACE ON DRIVE X CONTINUE (Y/N)?". A "Y" response will cause the Editor to continue; an "N" response will cause the Editor to return control to MDOS (refer to Table 2-1). Once the Editor has been invoked, it will display a message indicating that the file, FILENAME1, was opened or created new, and whether the file is line-numbered (with line numbers).

#### CAUTIONS

DO NOT REMOVE ANY DISKETTES FROM THE SYSTEM WHILE EDITORM IS IN OPERATION DUE TO POSSIBLE SCRATCH FILES ON EITHER DRIVE.

DO NOT INVOKE EDITORM WHEN ANY NON-MDOS DISKETTES (i.e., COBOL FMS) ARE IN ANY DRIVES, AS THEY MAY BE DESTROYED.

### 1.6 MODES OF OPERATION

Two basic modes of operation are available with the Editor: CRT-MODE and SCROLL-MODE. They determine the type of editing and the command that may be used. (See Figure 1-1).



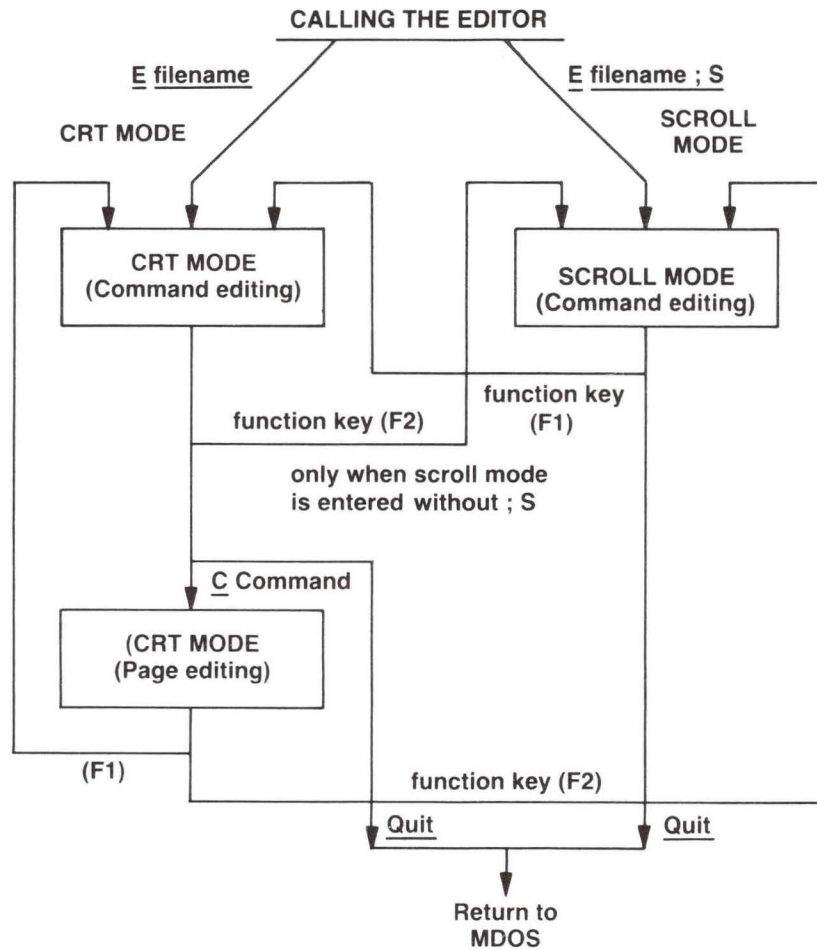


FIGURE 1-1. Entering CRT Editor from MDOS

### 1.6.1 CRT-MODE

The CRT-MODE supports both page editing (also known as screen editing) and command editing. When the Editor is used to edit an existing file, it dynamically adapts to command editing (refer to Figure 1-2). With command editing, the user enters edit commands (Table 1-1) at the cursor position following the prompt (>) on line 22, while continually viewing a portion of the file on the CRT screen (lines 1 through 20). Thus, the screen becomes a window where the portion of the file currently acted upon is displayed. The Editor prompt in this window (column 1) serves as a pointer to indicate the current file position. Table 1-2 lists the special character commands that are controlled by the use of the terminal's special-purpose keys when performing command editing. Thus, the Edit commands in Table 1-1 can themselves be edited, using the command edit functions shown in Table 1-2.

Page (screen) editing can be initiated only in the CRT-MODE. It is initiated when the Editor is invoked to create a new file (refer to Figure 1-3), or when one of the following commands is entered:

- CHANGE (without a STRING2)
- INSERT (for text files without line numbers)
- NUMBER (for source files with line numbers)

Page (screen) editing is terminated via a mode change request (use of the F1 or F2 function key).

With page (screen) editing, the user is positioned on the window and uses the CRT control characters to insert, change, or delete characters and/or lines within the file. Table 1-3 lists the special character commands that are controlled by use of the terminal's special purpose keys when performing page editing.

The Editor will accept only valid printable ASCII characters. Any nonprintable character received from the terminal or file in a data stream will be ignored. The EXORterm keyboard provides the means of inputting the full 128-character ASCII set, plus various function keys. The function keys F1 through F7 are used to select the special character commands in the CRT-MODE, as shown in Table 1-2. In this mode, the bottom two lines on the screen are used to display the mnemonic function name for each of the seven function keys.

```
>0010  NAM POLL
0020  OPT MEM
0030  PIA1AC EQU $4005
0040  PIA1BC EQU $4007
0050  PIA2AC EQU $4009
0060  PIA2BC EQU $400B
0070  ORG $100
0080  POLL LDA A PIA1AC
0090  BMI ROUT1
0100  ROL A
0110  BMI ROUT2
0120  LDA A PIS1BC
0130  BMI ROUT3
0140  ROL A
0150  BMI ROUT4
0160  LDA A PIA2AC
0170  BMI ROUT5
0180  ROL A
0190  BMI ROUT6
0200  LDA A PIA2BC
EDITING OLD FILE: LTR2 .SA:0 WITH LINE NUMBERS
>
F1      F2      F3      F4      F5      F6      F7
CRT     SCROLL PAGE ^ PAGE v LINE ^ LINE v DUP     LTR1     .SA:0
```

FIGURE 1-2. CRT-MODE (Command Editing)

TABLE 1-1. Edit Commands

COMMAND	DESCRIPTION
1-9999 (NOTE 1)	Insert, replace, or delete a line from the file.
C (CHANGE)	Change a string within a record (or a group of records).
DEL (DELETE)	Delete data from the file.
D (DUPLICATE)	Copy a group of records from one place within a file to another place within the same file.
E	Terminate the edit session. Begin editing another file.
EX (EXTEND)	Insert data at the end of records.
F, FIND	Find a string or a line.
I (INSERT)	Insert data into the file.
L, LIST	Display a portion of or the entire file.
MERG (MERGE)	Retrieve records from another MDOS file and include them in the edited file.
MOVE	Move a group of records from one place within a file to another place within the same file.
N, NUMBER	Insert data into the file and prompt with the next line number. (For editing files with line numbers.)
PRIN (PRINT)	Print data at an associated line printer.
QUIT	Terminate the edit session and return control to MDOS.
R, RANGE	Establish default values for the vertical and horizontal ranges (for LIST, PRINT, CHANGE, and FIND.)
RESE (RESEQUENCE)	Number, renumber, or unnumber the file.
SAVE (NOTE 2)	Save the edited file into the named file.
S (SEARCH)	Find the next occurrence of a string.
TAB	Alter the tab settings used for data input.
V (VERIFY)	Display at the terminal, anytime a change is made to the file, the values after the change.
X, XTRACT	Insert the data in the buffer into the file.
NOTE 1: Valid only for non-line numbered files.	
NOTE 2: Works only in non-line numbered mode for both line numbered and non-line numbered files.	

TABLE 1-2. Command Editing (CRT-MODE)

EXORTERM KEYS	DESCRIPTION
←	Move cursor to the left one column on line 22 with no character modification.
→	Move cursor to the right one column on line 22 with no character modification.
↑	Move display pointer up one line on the window display (lines 1 through 20), with page wraparound.
↓	Move display pointer down one line on the window display (lines 1 through 20), with page wraparound.
CTL-W	Pause the processing of this list. (CTRL and W keys)
CTL-X	Cancel this line. (CTRL and X keys)
DEL*	Delete previous character.
INS CHAR	All characters in this line, starting with the cursor column, are moved right one column. The cursor is then set at the inserted space.
DEL CHAR	All characters in the line, starting with the cursor column, are moved left one column. The cursor does not change position.
LINE* ERAS	All data from the cursor position to the end of the line is deleted.
F1	Stop page editing; start command editing.
F2	Change mode to SCROLL-MODE.
F3	Scroll screen forward one page.
F4	Scroll screen back one page.
F5	Scroll screen forward one line.
F6	Scroll screen back one line.
* denotes shift key function	

TABLE 1-3. Page (Screen) Editing (CRT-MODE)

EXOR TERM KEYS	DESCRIPTION
←	Move cursor to the left one column on the window display, with no character modification.
→	Move cursor to the right one column on the window display, with no character modification.
↑	Move display pointer and cursor up one line on the window display, with page wraparound.
↓	Move display pointer and cursor down one line on the window display, with page wraparound.
CTL-W	Pause the processing of this list. (CTRL and W keys)
CTL-X	Cancel this line. (CTRL and X keys)
←	Move the cursor left to previous tab stop.
→	Move cursor right to next tab stop.
DEL*	Delete previous character.
INS CHAR	All characters in this line, starting with the cursor column, are moved right one column. Line will be truncated if extended past column 79.
DEL CHAR	All characters in the line, starting to the right of the cursor, are moved one to the left. The cursor does not change position.
INS LINE	Insert a line at the current file position.
DEL LINE	Delete a line at the current file position
LINE* ERAS	All data from the cursor position to the end of the line is deleted.
F1	Stop page editing; start command editing.
F2	Change mode to SCROLL-MODE.
F3	Scroll screen forward one page.
F4	Scroll screen back one page.
F5	Scroll screen forward one line
F6	Scroll screen back one line.
F7	Duplicate (on this line) the character on line immediately above.

\* denotes shift key function

>0010

EDITING NEW FILE: LTR3 .SA:0 WITH LINE NUMBERS

>

F1	F2	F3	F4	F5	F6	F7		
CRT	SCROLL	PAGE ^	PAGE v	LINE ^	LINE v	DUP	LTR3	.SA:0

FIGURE 1-3. CRT-MODE (Page (Screen) Editing)

### 1.6.2 SCROLL-MODE

In the SCROLL-MODE, the user performs editing (Table 1-1) with each command line scrolling up and the resultant display on the next line. Table 1-4 lists the special character commands used in SCROLL-MODE. Figure 1-4 shows a typical program being edited in SCROLL-MODE.

TABLE 1-4. Command Editing (SCROLL-MODE)

EXORTERM KEYS	DESCRIPTION
CTL-W	Pause the processing of this list. (CTRL and W keys)
CTL-X	Cancel this line. (CTRL and X keys)
→	Echo spaces to move to the right to next tab stop.
DEL	Delete previous character and indicate so by echoing deleted character.
BREAK	Stop processing; await a new command.
F1	Change mode to CRT-MODE. (Not functional with the S option).

```

=E LTR3;S
MDOS EDITOR RELEASE 3.0
COPYRIGHT BY MOTOROLA 1978

EDITING OLD FILE: LTR3 .SA:0 WITH LINE NUMBERS
>L 10-160
0010  NAM POL
0020  OPT MEM
0030  PIA1AC EQU $3005
0040  PIA1BC EQU $4007
0050  PIA2AC EQU $4009
0060  PIA2BC EQU $400B
0070  ORG $100
0080  POLL LDA A PIA1AC
0090  BMI ROUT1
0100  ROL A
0110  BMI ROUT2
0120  LDA A PIA1BC
0130  BMI ROUT3
0140  ROL A
0150  BMI ROUT4
0160  LDA A PIA2AC

>

```

FIGURE 1-4. SCROLL-MODE (Command Editing)

## CHAPTER 2

### EDIT COMMANDS

#### 2.1 INTRODUCTION

There are two levels of edit features available: a basic set which the user may master in a relatively short period of time, and an advanced set which gives the user much more flexibility in editing. The advanced set may be mastered as needed.

The basic command set includes the 1-9999, CHANGE, DELETE, PRINT, SAVE, and QUIT commands, along with the elementary edit feature.

The editor prompts the user for the next command by displaying a greater-than sign (>). The command is entered, followed by a carriage return. Many of the commands have an abbreviated form, or alias, that the programmer may use if desired. Following are descriptions of the editor commands.

#### 2.2 COMMAND SYNTAX

Editor commands may be entered in either upper or lower case and in either a short or long form. Since only the first four characters of any command have significance, all commands may be abbreviated to the first four characters. All commands must be followed by a space. In all of the commands, the asterisk (\*) may be used to indicate the current line position when referring to a line number.

In defining the commands provided by the editor, the most common syntactical elements - vertical and horizontal range - are defined as follows:

VERTICAL RANGE = [[BEGINNING LINE NUMBER] [-ENDING LINE NUMBER]] OR  
[[COUNT1-]COUNT2]

WHERE: BEGINNING LINE NUMBER/ENDING LINE NUMBER apply to line-numbered files only. Valid line numbers are 1-9999.

COUNT1/COUNT2 apply to unlined files.

COUNT1 is the offset from the current line at which to begin the function performed. The offset may be negative for most commands. An asterisk (\*) denotes the current line.

COUNT2 is the number of lines upon which to perform the function.

When editing with line numbers, an illogical vertical range will result in a command syntax error.

The default for omitted values (where not overridden by the RANGE command) will result in the function being performed on the current line.



HORIZONTAL RANGE = [ :COLUMN1[-COLUMN2] ] OR [ :Fi[-Fi+n] ]

WHERE: COLUMN1 is the column on each line in which to begin the function to be performed.  
COLUMN2 is the column on each line in which to end the function to be performed.  
Fi is the field defined by the i'th tab stop through the i'th + 1 tab stop.

The default values where no horizontal range is specified will result in the function being performed on the entire line.

### 2.3 1-9999 COMMAND

FUNCTION: Insert, replace, or delete a line from the file. (This command applies to line-numbered files only.)

SYNTAX: n DATA

WHERE: n is the line number of the line to be inserted, replaced, or deleted. DATA is the new value of the line.

The entry of a line number with no data will imply a request to delete the line if it exists. If the line does not exist, it will cause a null line to be inserted. The entry of a line number with data will imply a request to replace that line if it exists. If it does not exist, the line number and data will be inserted into the proper location.

CRT-MODE: The line replaced or inserted, or the line immediately following the line deleted, is the new current line position.

#### EXAMPLES:

1. To insert line number 200 into the file (line 200 does not currently exist):  
>200 DATA ON LINE 200cr
2. To change line number 200 (line 200 already exists):  
>200 NEW DATA ON LINE 200cr
3. To delete line 200 from the file (line 200 already exists):  
>200cr
4. To insert a null line numbered 800 (line 800 does not currently exist):  
>800cr

For inserts or replacements, the current file position is moved to the line that was just inserted or replaced. For deletions, if the deleted line was not the last line in the file, the file position pointer (>) is moved to the line immediately after the deleted line. Otherwise, it is moved to the line immediately before.

5. To replace line number 240:  
>240 \* NEW LINE #240cr

## 2.4 CHANGE COMMAND

FUNCTION: Change a string within a record (or a group of records).

SYNTAX: CHANGE [VERTICAL RANGE] [HORIZONTAL RANGE]  
[.TRANSPARENT CHAR] [;COUNT] [STRING1] [STRING2] [REPETITION]

ALIAS: C, change, c

WHERE: STRING1 is the target string to be found and changed.

STRING2 is the value to which STRING1 is to be changed.

COUNT indicates that the COUNT'th occurrence of STRING1 in a line is the target string.

REPETITION indicates the number of lines on which STRING1 is to be found and changed.

TRANSPARENT CHAR is a character within STRING1 that is to be ignored when testing for equality to STRING1.

An "A" in the COUNT or REPETITION field indicates all occurrences. The default value for COUNT is 1 (the first occurrence), and the default value for REPETITION is all lines within the VERTICAL RANGE. String data is delimited by the first non-blank, non-numeric character. The same delimiter must also terminate STRING1 and STRING2 (if present). Care should be used in picking a string delimiter, because the use of "." or ";" when the TRANSPARENT CHAR or COUNT is not specified will cause unpredictable results. When the Change command is executed with no parameters, the next occurrence of the string data last specified by the user is modified. A null STRING1 will be interpreted to imply insertion of STRING2 at the beginning of the HORIZONTAL RANGE. A null STRING2 will be interpreted to imply deletion of STRING1. If STRING2 does not exist and the user is in the CRT-MODE, the user will begin page editing with the cursor resting on the first character of STRING1. If STRING2 does not exist and the user is in the SCROLL-MODE, the target line will be displayed to STRING1 and the user is placed in the input process (see the INSERT and NUMBER commands). When only COLUMN1 (or tab stop Fi) of the HORIZONTAL RANGE is specified and no string is specified, the following will occur. If in the CRT-MODE, the user will begin page editing with the cursor resting on the column position (or the field position) specified; if in the SCROLL-MODE, the target line will be displayed to COLUMN1 (or tab stop Fi) and the user will be placed in the input process (see the INSERT and NUMBER commands). In the CRT-MODE, a Change command with only a null STRING1 initiates page editing at the current line position. Underscore is the default transparent character.

If STRING1 is found, the current file position is moved to the last line of the VERTICAL RANGE where a change was made.

CRT-MODE: The last data line modified by the change is at the current position line.

EXAMPLES:

1. Change the string SAM to BILL on the current position line.

```
>C /SAM/BILL/cr
```

2. Change first occurrence of the string SAM to BILL on each line from line numbers 100 through 200.  
>C 100-200/SAM/BILL/cr
3. Change first occurrence of the string SAM (that occurs after column 30 and before column 60) to the string BILL on line number 100.  
>C 100:30-60/SAM/BILL/cr
4. Change third occurrence of the string SAM to BILL on the current position line.  
>C ;3/SAM/BILL/cr
5. Change all occurrences of the string SAM to BILL on each line from line numbers 100 through 200.  
>C 100-200;A/SAM/BILL/cr
6. Delete the string BOB from line number 120.  
>C 120/BOB//cr
7. Insert the string BOB at beginning of current line.  
>C //BOB/cr
8. Change the strings LABELx (where x can be any character or digit) to the string 1.25 on line numbers 100 through 500.  
>C 100-500.X/LABELX/1.25/cr
9. Change first occurrence of the string SAM to BILL on lines from line numbers 100 through 200. Then change the next occurrence of the string SAM to BILL from wherever it is to the end of the file.  
>C 100-200/SAM/BILL/lcr  
>Ccr
10. Change the second occurrence of the string LABxL (where x can be any character or digit) to the string LABEL where it occurs between columns 30 and 60 on line numbers 100 through 300.  
>C 100-300:30-60.X;2/LABxL/LABEL/cr
11. Change the string SAM to BILL on next five lines (where file being edited does not have line numbers).  
>C \*-5/SAM/BILL/cr
12. Begin page editing at the beginning of this line.  
>C //cr

13. Begin page editing at the beginning of the string SAM on this line.

>C /SAM/cr

14. Begin page editing on column 12 of line number 100.

>C 100:12/cr

15. Begin page editing on the second occurrence of the string X on line 200.

>C 200;2/X/cr

## 2.5 DELETE COMMAND

**FUNCTION:** Delete data from file and move current file position to line immediately after last line on which a deletion was performed.

**SYNTAX:** DELETE [VERTICAL RANGE] [HORIZONTAL RANGE]

**ALIAS:** DEL, delete, del

Specification of a HORIZONTAL RANGE will cause the data within the range to be deleted from each line within the VERTICAL RANGE.

**CRT-MODE:** The line immediately following the last line affected is the new current position line.

### EXAMPLES:

1. Delete the line at the current file position.

```
>DELcr
```

2. Delete lines 100 through 200 (line numbered files only).

```
>DEL 100-200cr
```

3. Delete data in columns 10 through 20 on line 210 (line numbered files only).

```
>DEL 210 :10-20cr
```

4. Delete line number 210 (line numbered files only).

```
>DEL 210cr
```

5. Delete next 5 lines (when editing a file without line numbers).

```
>DEL *-5cr
```

6. Delete next 3 lines, starting from 6th line before current file position (non-line numbered files only).

```
>DEL -6-3cr
```

7. Delete data in column 6 (first non-line number field column) in lines 30 through 100 (line numbered files only).

```
>DEL 30-100 :6-6cr
```

It should be noted that the HORIZONTAL RANGE is absolute - i.e., it includes the line number columns for line numbered files. Therefore, specifying a HORIZONTAL RANGE column of less than six (<6) will delete the line(s) specified instead of the column(s). Also, a HORIZONTAL RANGE must be shown, even if only one column is to be deleted - i.e., "n-n". A HORIZONTAL RANGE of just "n" implies to delete columns "n" through the end of the line. If a null line(s) results (n=1 for unlined file, n<7 for lined file), the line(s) will be deleted.

## 2.6 DUPLICATE COMMAND

**FUNCTION:** Copy a group of records from one place within a file to another place within the same file.

**SYNTAX:** DUPLICATE [VERTICAL RANGE] [/NEW LINE NUMBER[,NEW INCREMENT]]

**ALIAS:** D, duplicate, d

**WHERE:** NEW LINE NUMBER is the destination line number for the first line copied (line numbered files only).

NEW INCREMENT is the line number increment to be applied to each subsequent line of the copy (line numbered files only). Default equals 10.

The Duplicate command will copy the requested data into the XTRACT buffer area. (Any old data in the buffer is lost.) The data can then be moved from the buffer via the XTRACT command as many times as needed. When used with line numbered files specifying the NEW LINE NUMBER, the Duplicate command will automatically invoke the XTRACT command once to copy the data from the buffer to the target area. See the BLOCK/LINE MERGE RULES, paragraph 2.24, for the treatment of line number conflicts. When files without line numbers are being edited, the pointer must be moved to the desired location, after which the XTRACT command should be invoked.

**CRT-MODE:** If the NEW LINE NUMBER was not utilized, the screen is unchanged. If it was specified, the current position line contains the line following the last line copied at its new location.

### EXAMPLES:

1. Duplicate this line (put this line into the XTRACT buffer). Previous contents of XTRACT buffer are lost.

>Dcr

2. Duplicate lines 100 through 200.

>D 100-200cr

3. Duplicate next 10 lines (when editing a file without line numbers).

>D \*-10cr

4. Duplicate this line and put it at line number 111. Old line 111 becomes 112 when resequenced.

>D /111cr

5. Duplicate line numbers 110 through 130 and put them at line number 151, with an increment of 2 between lines.

>D 110-130/151,2cr

6. Duplicate line 130 and put it at 143.

>D 130/143cr

## 2.7 E COMMAND

**FUNCTION:** An editor command that will terminate this edit session and initiate the editing of another named file. (For command specifics, see INVOKING THE EDITOR, paragraph 1.5). Note that MDOS file names and command options currently must be upper case only.

**SYNTAX:** E <filename>[;<options>]

**ALIAS:** e

**WHERE:** <filename> is a proper MDOS file name, including drive number, and <options> are any valid EDITORM options (see paragraph 1.5).

## 2.8 EXTEND COMMAND

FUNCTION: Insert data at the end of records.

SYNTAX: EXTEND [VERTICAL RANGE]STRING

ALIAS: EX, extend

WHERE: STRING is the value to be appended to the end of the records in the VERTICAL RANGE.

The STRING must be delimited by the first non-blank non-numeric character. The same delimiter must be used to terminate the string.

CRT-MODE: The current position line is the last line modified by the Extend command.

### EXAMPLES:

1. Put a comma at the end of the current line.

```
>EX /,/cr
```

2. Put a period at the end of line 100 (line numbered files only).

```
>EX 100/./cr
```

3. Put a period at the end of all lines 100 through 200, and then move current file position to line 200.

```
>EX 100-200/./cr
```

4. Put a period at the end of each of the six lines starting at 15 lines before the current file position, and then move the current file position to the last line extended (6th line) (non-line numbered files only).

```
>EX -15-6/./cr
```



## 2.9 FIND COMMAND

**FUNCTION:** Find a string or a line.

**SYNTAX:** FIND [VERTICAL RANGE] [HORIZONTAL RANGE] [.TRANSPARENT CHAR] [;COUNT]  
STRING [REPETITION]

**ALIAS:** F, find, f

**WHERE:** TRANSPARENT CHAR is a character within the STRING that is to be ignored in determining search satisfaction.

STRING is the string to be found.

COUNT indicates that the COUNT'th occurrence of the STRING in a line is the target string.

REPETITION indicates the number of times -1 the STRING is to be found, except for REPETITION=1, which is unaltered.

The STRING must be delimited by the first non-blank, non-numeric character. The same delimiter must be used to terminate the string. When a FIND command is executed with no parameters, it implies a request to re-execute the last FIND command, starting from the current file position. When the FIND command is executed with only a VERTICAL RANGE specified, it implies a request to position the editor at the start of the VERTICAL RANGE. FIND 0 will imply a request to position the editor at the beginning of the file and, conversely, FIND 9999 will imply a request to position the editor at the end of the file. When the VERTICAL RANGE is not specified, the string search will begin at the beginning of the file. An A in the COUNT or REPETITION field will imply a request to find and display all the occurrences of the STRING. The default value for COUNT is 1, and the default value for REPETITION is 1. When the REPETITION is greater than 1 in the CRT-MODE, the screen will display all but the last of the lines with an occurrence of the STRING. If the screen gets full, the display will scroll up until the list is exhausted or the user types CTL-W (to pause the list) or BREAK (to abort the list). The screen may be returned to the normal CRT-MODE display by entering a null command or any other valid editor command. Underscore (   ) is the default transparent character.

Upon completion of a successful FIND command, the Editor is positioned to the line containing the last encounter of the STRING unless all occurrences (A) was requested, which positions the Editor to the line after the last occurrence.

When a string is not found, the current file position is moved to the BEGINNING LINE NUMBER or COUNT1 of the VERTICAL RANGE.

### EXAMPLES:

1. Move the current file position to the beginning of the file. NOTE: If the file contains more than 22 lines, the file position is set to the 2nd line of the file.

>F 0cr

2. Move current file position to the 9th line before current line.

>F -9cr

3. Move current file position to the 5th line after current line.  
>F 5cr
4. Move current file position to line 140 (line numbered files only).  
>F 140cr
5. Move current file position to end of the file.  
>F 9999cr
6. Move current file position to line with the first occurrence of the string SAM.  
>F /SAM/cr
7. Move current file position to first line with two occurrences of the string SAM, between lines 100 and 200 (line numbered files only).  
>F 100-200;2/SAM/cr
8. List all occurrences of the string SAM, then move current file position to line after last occurrence-line.  
>F /SAM/cr
9. List all lines with 2 or more occurrences of the string SAM, then move current file position to line after last occurrence-line.  
>F ;2/SAM/cr
10. List all occurrences, within the 70 lines starting from 50 lines before current file position, of the string ACTIVITY, then move current file position to line after last occurrence-line (non-line numbered files only).  
>F -50-70/ACTIVITY/Acr

## 2.10 INSERT COMMAND

**FUNCTION:** Insert data into file but do not alter current file position (for editing files without line numbers only).

**SYNTAX:** INSERT

**ALIAS:** I, insert, i

**SCROLL-MODE:** The Insert command initiates the input process in which all data following the command is inserted into the file until a null line is encountered - i.e., two consecutive carriage returns. The null line is not inserted. To insert a blank line, type at least one blank (space) character before typing a cr.

**CRT-MODE:** The Insert command causes entry in the page editing (CRT) mode, inserts a blank line immediately in front of the cursor, and then positions the cursor at the beginning of the new blank line.

### EXAMPLES:

1. To insert data into a file before the current file position in SCROLL-MODE:

```
>Icr  
>THIS DATA IScr  
>NEW LINES THATcr  
>ARE TO BE ENTEREDcr  
>INTO THE FILEcr  
>cr
```

2. To insert data into a file before the current file position in CRT-MODE (Command Editing):

```
>Icr      (enters Page Editing)
```

## 2.11 LIST COMMAND

**FUNCTION:** Display a portion of the file but do not alter the current file position.

**SYNTAX:** LIST [VERTICAL RANGE]

**ALIAS:** L, list, l

The List command will cause the data within the VERTICAL RANGE to be displayed on the screen. If the list is too long for the screen, the display will be scrolled up until the list is exhausted or the user types CTL-W (to pause the list) or BREAK (to abort the list).

### EXAMPLES:

1. Display entire file, starting from first line.

>LCR

2. Display only the current line.

>L \*cr or >L 0cr

(NOTE: >L 0cr is for non-line number files only.)

3. Display line numbers 100 through 200 (line numbered files only).

>L 100-200cr

4. Display next 10 lines (non-line numbered files only), starting from current line.

>L \*-10cr or >L 0-10cr

5. Display next 7 lines, starting from 12th line after current line (non-line numbered files only.)

>L 12-7cr

6. Display next 13 lines, starting from 12th line after current line (non-line numbered files only).

>L -4-13cr

## 2.12 MERGE COMMAND

**FUNCTION:** Retrieve records from another MDOS file and include them in the edited file.

**SYNTAX:** MERGE [VERTICAL RANGE] FILENAME [NEW LINE NUMBER[,NEW INCREMENT]]

**ALIAS:** MERG, merge, merg

**WHERE:** FILENAME specifies a valid MDOS file.

VERTICAL RANGE specifies the range in FILENAME to be copied into the edited file.

NEW LINE NUMBER is the line number to be assigned to the first line from FILENAME when it is inserted into the file.

NEW INCREMENT is the increment to be added to each successive line number as the lines are inserted into the edited file.

If the VERTICAL RANGE is not specified, it will imply the request to get the entire FILENAME. If the user is editing without line numbers, the data from FILENAME will be inserted before the current position line. The line numbers that may be in FILENAME are lost upon extract. The line numbers applied will be as specified via NEW LINE NUMBER and NEW INCREMENT, if specified, or will default to the current position line plus 10 with an increment of 10. If the establishment of new line numbers causes a conflict, the BLOCK/LINE MERGE RULES given in paragraph 2.24 will apply.

The VERTICAL RANGE, if used, and response to the question prompt "IS THE SOURCE FILE LINE NUMBERED (Y/N)?" must correspond to the line attribute of FILENAME. In non-line numbered files, COUNT1 (of the VERTICAL RANGE) is the start line number in FILENAME at which to start copying, and COUNT2 is the number of lines to be copied.

### EXAMPLES:

1. Insert all lines in file named SAM into edited file before current file location.

```
>MERG SAMcr
```

2. Starting with line number 400, insert all lines in file named SAM into edited file, with an increment of 1 between line numbers.

```
>MERG SAM 400,1cr
```

3. Insert line numbers 100 through 300 in file named SAM into edited file, beginning with line number 500 and with increments of 2 between lines.

```
>MERG 100-300 SAM 500,2cr
```

## 2.13 MOVE COMMAND

FUNCTION: Move a group of records from one place to another within the file.

SYNTAX: MOVE [VERTICAL RANGE] [/NEW LINE NUMBER],NEW INCREMENT]]

ALIAS: move

WHERE: NEW LINE NUMBER is the destination line number for first line moved (line-numbered files only).

NEW INCREMENT is the line number increment to be applied to each subsequent line of the move. Default = 10. (Line numbered files only.)

The Move command generates a Duplicate command followed by a Delete command. For additional information, see those command descriptions.

### EXAMPLES:

1. Move this line to the XTRACT buffer. Previous contents of XTRACT buffer are lost.

```
>MOVEcr
```

2. Move line number 100 to XTRACT buffer and move current file position to next line.

```
>MOVE 100cr
```

3. Move line number 100 through 200 into XTRACT buffer.

```
>MOVE 100-200cr
```

4. Move next 10 lines into XTRACT buffer (when editing a file without line numbers) and move current file position to line immediately after 10th line.

```
>MOVE *-10cr
```

5. Move line at current file position to line number 131.

```
>MOVE /131cr
```

6. Move line 120 to 123.

```
>MOVE 120/123cr
```

7. Move lines 150 through 200 to line 190, with an increment of 1 between lines.

```
>MOVE 150-200/190,1cr
```

## 2.14 NUMBER COMMAND

**FUNCTION:** Insert data into the file and prompt user with the next line number for insert (for editing line-numbered files only).

**SYNTAX:** NUMBER [NEW LINE NUMBER [,NEW INCREMENT]]

**ALIAS:** N, number, n

**WHERE:** NEW LINE NUMBER is the first line number to be prompted.

NEW INCREMENT is the value to be added to the NEW LINE NUMBER to form each succeeding line number prompted.

The Number command initiates the input process in which all data following the command is inserted into the file. With no parameters, the Number command will cause the Editor to form the line number prompt by adding the default increment of 10 to the last line number in the file. Prompting will continue after each carriage return until a null line is encountered - i.e., two consecutive carriage returns.

If any line number generated matches an existing line number, the existing line will be replaced with the new line.

If NEW INCREMENT is specified, then NEW LINE NUMBER must also be specified. To insert a blank line, type at least one blank (space) character before a carriage return.

### EXAMPLES:

1. Begin page editing and prompting line numbers at end of file in increments of 10.

```
>Ncr
```

2. Begin page editing at line number 100, with increments of 10.

```
>N 100cr
```

3. Begin page editing at line number 200, with line number prompts in increments of 1.

```
>N 100,1cr
```

4. Insert 3 lines between line numbers 40 and 50 in the SCROLL-MODE, with an increment of 2.

```
>N 42,2cr
```

```
0042 THESE 3 LINES ARE INSERTEDcr
```

```
0044 THE LINE NUMBERS AREcr
```

```
0046 AUTOMATICALLY PROMPTED.cr
```

```
>
```

## 2.15 PRINT COMMAND

**FUNCTION:** Print a portion of or the entire file on a printer but do not alter current file position. The printer is adjusted to start on the top of a page and restore the paper to the same relative starting point.

**SYNTAX:** PRINT [VERTICAL RANGE] [SPACING]

**ALIAS:** PRIN, print, prin

**WHERE:** SPACING is "D" to indicate double spacing and "T" to indicate triple spacing.

### EXAMPLES:

1. Print entire file, starting from first line.

```
>PRINcr
```

2. Print current line only.

```
>PRIN *cr or >PRIN 0cr (non-line numbered files only).
```

3. Print line numbers 100 through 200 with double spacing (line numbered files only).

```
>PRIN 100-200 Dcr
```

4. Print next 10 lines (when editing a file without line numbers) with triple spacing.

```
>PRIN *-10 Tcr
```



## 2.16 QUIT COMMAND

FUNCTION: Terminate the edit and return control to MDOS.

SYNTAX: QUIT [A]

ALIAS: quit

WHERE: A implies abort the edit.

Unless the A option is specified, the Quit command causes the results of the edit to be saved prior to termination.

### EXAMPLES:

1. Terminate the edit and save the results.

```
>QUITcr
```

2. Abort the edit and do not modify the source file, but delete all scratch (EDIT intermediate) files.

```
>QUIT Acr
```

## 2.17 RANGE COMMAND

FUNCTION: Establish a default value for the vertical and horizontal ranges.

SYNTAX: RANGE [VERTICAL RANGE] [HORIZONTAL RANGE]

ALIAS: R, range, r

With no parameters, the Range command deletes the previously established ranges. The Range command provides the default ranges for LIST, PRINT, CHANGE, and FIND. For non-line numbered files, COUNT1 of VERTICAL RANGE cannot be negative.

The default HORIZONTAL RANGE should not be set >79 for EXORterm terminals, nor greater than the maximum printing column number for non-EXORterm terminals.

### EXAMPLES:

1. Set default VERTICAL RANGE to lines 100 through 300 (line numbered files only).

```
>R 100-300cr
```

2. Set default HORIZONTAL RANGE to edit a full 132 character line.

```
>R :1-132cr
```

3. Set VERTICAL RANGE default to lines 100 through 200, and set HORIZONTAL RANGE to columns 16 through 80 (line numbered files only).

```
>R 100-200 :16-80cr
```

## 2.18 RESEQUENCE COMMAND

**FUNCTION:** Number, renumber, or unnumber the file, but do not alter current file position.

**SYNTAX:** RESEQUENCE [BEGINNING LINE NUMBER] [/NEW LINE NUMBER  
[,NEW INCREMENT]] [N]

**ALIAS:** RESE, resequence, rese

**WHERE:** BEGINNING LINE NUMBER is the line number to be assigned to the first line resequenced.

NEW LINE NUMBER is the line number to be assigned to the first line resequenced.

NEW INCREMENT is the increment to be applied to form each subsequent line number within the file.

N implies a request to strip the line numbers from the file.

If BEGINNING LINE NUMBER is not specified, the entire file will be resequenced. If NEW LINE NUMBER is not specified, resequencing will begin with the BEGINNING LINE NUMBER being changed to the {number of the line before it + new increment} and then all subsequent lines thereafter incremented by NEW INCREMENT. If NEW INCREMENT is not specified, it defaults to 10. The N option is mutually exclusive to the other options in this command.

### EXAMPLES:

1. Resequence the entire file. Begin new line numbers with 1, with increment of 1 between.

```
>RESE /1,1cr
```

2. Resequence entire file. Begin new line numbers with 100, with increment of 2 between line numbers.

```
>RESE /100,2cr
```

3. Resequence file beginning with line 13 being changed to 12 and all subsequent lines incremented to 1 (from new 12 onward).

```
>RESE 13/12,1cr
```

## 2.19 SAVE COMMAND for vertical range count (i.e., \*-9999, etc.)

This command works for both line numbered and non-line numbered files. Line numbered lines are saved as data.

FUNCTION: Save the edited file into the named file.

SYNTAX: SAVE [[VERTICAL RANGE] FILENAME]

ALIAS: save

WHERE: FILENAME implies that an extract function is to be performed. The referenced data (as modified by the edit) is written into the new file: FILENAME.

FILENAME must not already exist as an MDOS file and must be upper case only. If no VERTICAL RANGE is specified, the current modified version of the file is written into FILENAME. When FILENAME is not specified, the current modified version of the source file will replace the old version, and the old version is deleted from MDOS. The new version is then re-opened and repositioned to the beginning of the file for further editing.

### EXAMPLES:

1. Save the edited file to disk.

```
>SAVEcr
```

2. Save the edited file into a new file named SAM; current file position is not altered.

```
>SAVE SAMcr
```

3. Extract the next 20 lines and put them into a new file named SAM; current file position is not altered (non-line numbered files only).

```
>SAVE *-20 SAMcr
```

## 2.20 SEARCH COMMAND

**FUNCTION:** Search for the next occurrence of a string. If not found, move current file position to line immediately after current line.

**SYNTAX:** SEARCH [STRING] [A]

**ALIAS:** S, search, s

**WHERE:** STRING is the string to be found.

The STRING is delimited by the first non-blank character. The same delimiter must be used to terminate the string. The Search command will begin a string search at the line following the current line position. An A after the ending string delimiter implies a request to search for and list all occurrences of the STRING. A Search command with no STRING will search for the next occurrence of the last string specified by a Search command or a Find command. Thus, the Search command is equivalent to a Find 1-9999/STRING/ in an unlined file, or FIND n-9999/STRING/ in a numbered file (where n is the current line number +1).

Upon completion of a successful Search command, the Editor is positioned to the line containing the last successful encounter of the STRING, unless all occurrences (A) were requested, which positions to the line after the last occurrence.

### EXAMPLES:

1. Find the next occurrence of the string SAM.

```
>S /SAM/cr
```

2. Find the first occurrence of the string BILL and then find the next occurrence.

```
>F /BILL/cr
```

```
>Scr
```

3. Search for and display all occurrences of the string FRED.

```
>S /FRED/Acr
```

## 2.21 TAB COMMAND

**FUNCTION:** Alter the tab settings used for data input.

**SYNTAX:** TAB [CHAR] [COUNTN] 1-20

**ALIAS:** Tab

**WHERE:** CHAR is the non-numeric, non-space displayable character to designate a tab request during page editing. (The EXORterm tab keys may also be used.)

COUNTN is the column position to which tab is to be set.

When no CHAR is specified, the TAB function will not work. The tabs can be reset by entering the Tab command with no parameters. The 1 to 20 COUNTN's specified must be separated by commas. The COUNTN values will add additional tab stops until the tabs are reset by a Tab command with no parameters.

### EXAMPLES:

1. Set tabs at columns 10 and 20 and use the character Z as a special tab key.

```
>TAB Z 10,20cr
```

2. Add an additional tab stop at column 15.

```
>TAB 15cr
```

3. Delete all tab stops.

```
>TABcr
```

## 2.22 VERIFY COMMAND

**FUNCTION:** Display at the terminal, anytime a change is made to the file, the value after the change.

**SYNTAX:** VERIFY [OFF]

**ALIAS:** V, verify, v

**WHERE:** OFF disables the verify.

The verify is enabled by entering the Verify command with no parameters. With the verify enabled in the CRT-MODE, certain multiple update commands will display a list of occurrences rather than a sequential portion of the file surrounding the changed record. Should the list become too long for a single screen, the display will scroll up until the list is exhausted or the user types CTL-W (to pause the list) or BREAK (to abort the list). The screen can be returned to the normal display by entering a null command or any other valid Editor command. Use of the BREAK key will cause the update command to stop processing as of the last line listed on the screen.

**CRT-MODE:** Execution of the Verify command does not alter the display. The Verify command causes other update commands to display as noted above.

### EXAMPLES:

1. Enable verify.

```
>Vcr
```

2. Disable verify.

```
>V OFFcr
```

## 2.23 XTRACT COMMAND (see DUPLICATE and MOVE commands)

FUNCTION: Insert the data in the buffer into the file.

SYNTAX: XTRACT [NEW LINE NUMBER [,NEW INCREMENT]]

ALIAS: X, xtract, x

WHERE: NEW LINE NUMBER is the point at which to insert the first line from the buffer.

NEW INCREMENT is the increment to be applied to each subsequent line number inserted into the file. Default = 10.

In non-line numbered files, the Xtract command will always copy the data in the buffer into the file before the current position line. If the establishment of new line numbers generates a conflict, the BLOCK/LINE MERGE RULES (paragraph 2.24) will be applied to resolve the conflict.

### EXAMPLES:

1. Insert the data in the XTRACT buffer into the file before the current file position.

>Xcr

2. Insert the data in the XTRACT buffer into the file beginning with line number 100, with an increment of 10 between lines (line numbered files only).

>X 100cr

3. Insert the data in the XTRACT buffer into the file beginning with line number 200, with an increment of 1 between lines (line numbered files only).

>X 200, 1cr



## 2.24 BLOCK/LINE MERGE RULES

When the establishment of new line numbers for a block insert generates a conflict (a potential duplication or overlap of line numbers), the editor will display:

LINE NUMBER CONFLICT AT XXXX RESEQUENCE (Y/N)?

If the user responds with Y, the balance of the file will be resequenced only until there is no further conflict. If the user responds with N, the duplicated lines in the file will be overlaid with the new lines. Consider the following example:

User file prior to any editing:

```
010 1st line
020 2nd line
030 3rd line
040 4th line
050 5th line
060 6th line
070 7th line
080 8th line
090 9th line
100 10th line
```

User enters: DUPLICATE 10-50/74,4

The Editor will begin processing:

```
010 1st line
020 2nd line
030 3rd line
040 4th line
050 5th line
060 6th line
070 7th line
074 1st line
078 2nd line
```

LINE NUMBER CONFLICT AT 80 RESEQUENCE (Y/N)?

If the user types Y, the rest of the file will be:

```
082 3rd line
086 4th line
090 5th line
091 8th line
092 9th line
100 10th line
```

If the user types N, the rest of the file will be:

```
080 8th line
082 3rd line
086 4th line
090 5th line
100 10th line
```

## 2.25 EDITOR MESSAGES

Table 2-1 contains a list of the operator messages that may be encountered when using the Editor.

TABLE 2-1. Operator Messages

MESSAGE	PROBABLE CAUSE
EDITOR OVERLAY NOT AVAILABLE	All Editor overlay files must be in drive 0 throughout the edit session. Fatal error.
**02 NAME REQUIRED	The Editor was invoked without specifying the file to be edited.
**25 INVALID FILE NAME	The name of the file specified is not a valid MDOS file name.
**31 INVALID DEVICE	The device specified is not a valid MDOS device.
**05 XXXXXXXX.XX DUPLICATE FILE NAME	The file name specified cannot be created as it already exists.
**27 XXXXXXXX.XX IS WRITE PROTECTED	The file name specified has been marked write protected and cannot be edited.
**10 XXXXXXXX.XX HAS INVALID FILE TYPE	The file name specified is not an ASCII file and cannot be edited.
*07 OPTION CONFLICT	The options specified are conflicting
INSUFFICIENT DISK SPACE ON DRIVE X CONTINUE (Y/N)?	The system compared the free space on drive X to the file size of the file to be edited. If drive X is the drive on which the edited result is to be stored, a DISK SPACE FULL message will appear and the edit will be aborted when the file is written to disk. If drive X is not the drive on which the edited result is to be stored, the edit may continue as long as it makes only one pass through the file. An attempt to reposition the Editor to the beginning of the file may result in the disk space full error. If the DRIVE specified is the same drive on which the edited result is to be stored, this message is a caution to do a one-pass edit. A response of Y will allow the edit to continue. A response of N will return control to MDOS.

TABLE 2-1. Operator Messages (cont'd)

MESSAGE	PROBABLE CAUSE
SOURCE FILE SEQUENCE ERROR. RESEQUENCE REQUIRED	The source file contains non-numeric data in columns 1-4 or does not contain a space in column 5; or the line numbers in the source file are not in ascending order; or the line numbers in the source file are not unique.
WHAT? An invalid edit command was entered.	
COMMAND SYNTAX ERROR	A valid edit command was entered but the syntax of the command was not correct.
DEVICE NOT READY	A MERGE or PRINT command was received and the appropriate device is not ready.
**03 XXXXXXXX.XX DOES NOT EXIST	A MERGE command was received but the specified FILENAME does not exist.
COMMAND ABORTED	The BREAK key was repressed during file access, and the command last entered has been aborted prior to completion.
STRING NOT FOUND	The string specified in the CHANGE, FIND, or SEARCH command cannot be found in the specified ranges.
INVALID FIELD	The field specified within the HORIZONTAL RANGE has not been defined via tab stops.
LINE NUMBER OVERFLOW RESEQUENCE	An attempt to generate a line number resulted in a value greater than 9999, and wrapped around to or beyond 0000. The file must be resequenced or the command results are unpredictable.
XTRACT BUFFER OVERFLOW	An attempt to move or duplicate too large a block has occurred.
SOURCE LINES NOT FOUND	An attempt to MERGE, MOVE, DUPLICATE, or XTRACT occurred, and the referenced data does not exist; or the xtract buffer is empty.

TABLE 2-1. Operator Messages (cont'd)

MESSAGE	PROBABLE CAUSE
BOF OR EOF ENCOUNTERED	An attempt has occurred to page or scroll beyond the beginning or the end of the file.
LINE NUMBER CONFLICT AT XXXX RESEQUENCE (Y/N)?	Block editing has resulted in a line number conflict. The Editor will resolve the conflict according to the rules described in paragraph 2.10.
IS THE SOURCE FILE LINE NUMBERED (Y/N)?	A MERGE command was entered and the Editor needs to know whether or not to strip the first five characters of each record extracted.
**22 BUFFER OVERFLOW	A Change command or INS CHAR may cause the line length to exceed 132 characters which, in turn, will abort the command.



APPENDIX A

EDITING IN THE CRT-MODE AND ASSEMBLY OF PROGRAMS

## EDITING AND ASSEMBLING PROGRAMS

This appendix assumes that an EXORterm 150, 155, 200, or 220 is to be used as the communications terminal, and that the user has the M6800 Macro Assembler, M68MASR. If any other terminal is to be used, refer to SCROLL MODE EDITING, Appendix B.

If system is not active,

Also assumes EXORciser II

Activate system

. Call MDOS

Type MDOS following EXbug prompt (\*) and Executive Map prompt (E)

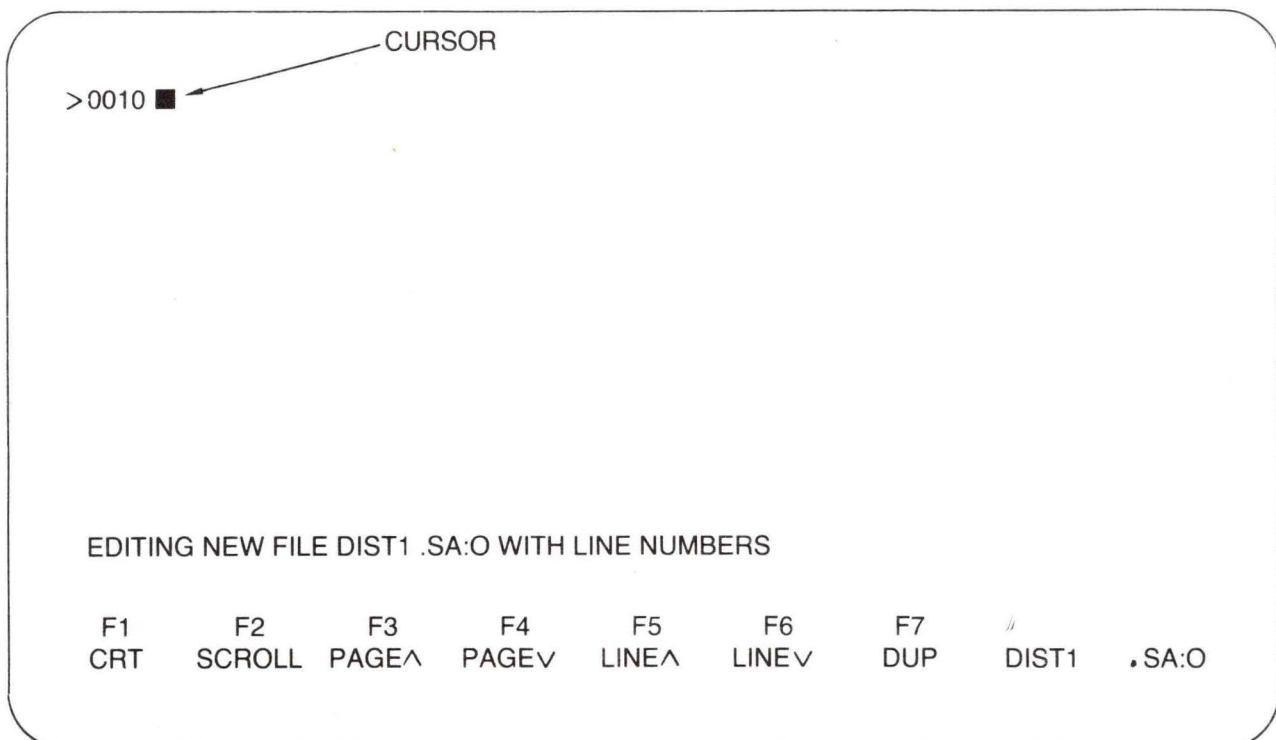
```
*E MDOS(cr)
MDOS 3.xx
=
```

. EDITING

Type EDIST1 following MDOS prompt (=)

```
=E DIST1(cr)
MDOS EDITOR RELEASE 3.xx
COPYRIGHT BY MOTOROLA 1979
```

The CRT Editor has been called and the following should appear on the CRT screen.



PROGRAM 1

Type into the Editor buffer the following, exactly as shown.

NOTE

The delete key may be used to correct any typing errors as they occur. Also, if a printer is not part of the system, change Line 20 to OPT S, NOP.

```
0010  NAM DIST1      (CR)
0020  OPT S         (CR)
0030  ORG $2000    (CR)
0040  LDX #$3000   (CR)
0050  CLRA         (CR)
0060  LDAB #10     (CR)
0070  AGAIN STAA 0,X (CR)
0080  INX          (CR)
0090  DECB        (CR)
0100  BNE AGAIN    (CR)
0110  BRA *        (CR)
>0120  END ■
```

EDITING NEW FILE DIST1 .SA:O WITH LINE NUMBERS

F1	F2	F3	F4	F5	F6	F7		
CRT	SCROLL	PAGE^	PAGEV	LINE^	LINEV	DUP	DIST1	SA:O

To terminate data entry,

Depress F1 Key

To terminate the Edit operation and save the file (DIST1.SA) on the diskette,

Type QUIT(cr).

The MDOS prompt (=) should now be displayed.



ASSEMBLING A PROGRAM

Type:

=RASM DIST1;AL(cr)  
M6800 MACROASSEMBLER 3.xx  
COPYRIGHT BY MOTOROLA 1979

This calls the macro Assembler and directs the listing to the printer (L option). The A option saves a file on the diskette that can be loaded into memory and executed. For a detailed discussion of macro assembler, refer to the Macro Assembler manual, M68MASR(D2).

NOTE

If a printer is not to be used, replace L with L=#CN.  
This will direct the assembly listing to the CRT.

The listing on the printer should be as follows:

---

PAGE 001 DIST1 .SA:0 DIST1

00010	00001					NAM	DIST1
00020	00002					OPT	S
00030	00003A	2000				ORG	\$2000
00040	00004A	2000	CE	3000	A	LDX	#\$3000
00050	00005A	2003	4F			CLRA	
00060	00006A	2004	C6	0A	A	LDAB	#10
00070	00007A	2006	A7	00	A AGAIN	STAA	0,X
00080	00008A	2008	08			INX	
00090	00009A	2009	5A			DECB	
00100	00010A	200A	26	FA	2006	BNE	AGAIN
00110	00011A	200C	20	FE	200C	BRA	*
00120	00012					END	
TOTAL ERRORS 00000 — 00000							

AGAIN 2006

The MDOS prompt (=) should be displayed.

There should be 0 errors. If this is not the case, then typing errors have occurred. These can be corrected later. Please continue.

PROGRAM 2

Type:

=E DIST2(cr)  
MDOS EDITOR RELEASE 3.xx  
COPYRIGHT BY MOTOROLA 1979

Enter the following program exactly as shown.  
(If not using a printer, change line 20 to OPT M,S,NOP)

```
>0010  NAM DIST2      (CR)
0020  OPT M,S        (CR)
0030  ORG $2020      (CR)
0040  LDAA #$AA      (CR)
0050  LDBB #10       (CR)
0060  LDX #$2000     (CR)
0070  STAA 0,X       (CR)
0080  MORE DECB      (CR)
0090  BNE MORE       (CR)
0100  END ■
```

EDITING NEW FILE DIST 2 .SA:0 WITH LINE NUMBERS

F1	F2	F3	F4	F5	F6	F7		
CRT	SCROLL	PAGE ^	PAGE v	LINE ^	LINE v	DUP	DIST2	SA:0

Depress F1 Key

Type QUIT(cr) .

The MDOS prompt (=) should appear.

ASSEMBLE PROGRAM 2

Type:

=RASM DIST1;L(cr) (If not using a printer, use "L=#CN" in place of "L")  
M6800 MACROASSEMBLER 3.xx  
COPYRIGHT BY MOTOROLA 1979

The listing from the printer should be as follows:

---

PAGE 001 DIST2 .SA:0 DIST2

```
00010 00001          NAM      DIST2
00020 00002          OPT      M,S
00030 00003A 2020    ORG      $2020
ERROR          235 — 00000
          00004A 2020 86 AA    A 0040    LDAA #$AA
ERROR          207 — 00004
          00005A 2022 00 0000 A 0050    LDBB #10
00060 00006A 2025 CE 2000 A          LDX      #$2000
00070 00007A 2028 A7 00    A          STAA     0,X
00080 00008A 202A 5A          MORE    DECB
00090 00009A 202B 26 FD 202A    BNE     MORE
00100 00010          END
TOTAL ERRORS 00002 — 00005
```

MORE 202A

There should be only two errors. If there are more, they were generated by the user.

- Error 235 is caused by the M in OPT M,S - the M should be removed.
- Error 207 is caused by the first B in LDBB - it should be LDAB.

To correct the above errors, the CRT EDITOR must be invoked again.

RE-EDITING PROGRAM 2

Type:

=E DIST2(cr)  
MDOS EDITOR RELEASE 3.xx  
COPYRIGHT BY MOTOROLA 1979

The following should be displayed on the CRT screen:

```
>0010  NAM  DIST2
0020  OPT  M,S
0030  ORG  $2020
0040  LDAA # $AA
0050  LDBB #10
0060  LDX  # $2000
0070  STAA 0,X
0080 MORE DECB
0090  BNE  MORE
0100  END
```

EDITING OLD FILE: DIST2 .SA:0 WITH LINE NUMBERS

> ■

F1	F2	F3	F4	F5	F6	F7		
CRT	SCROLL	PAGE ^	PAGE v	LINE ^	LINE v	DUP	DIST2	SA:0

To make the first correction,

Type C(cr) following the Editor prompt (>)

```
>0010 ■ NAM DIST2
0020  OPT M,S
0030  ORG $2020
0040  LDAA # $AA
0050  LDBB #10
0060  LDX # $2000
0070  STAA 0,X
0080 MORE DECB
0090  BNE MORE
0100  END
```

> C(CR)

F1	F2	F3	F4	F5	F6	F7		
CRT	SCROLL	PAGE ^	PAGE v	LINE ^	LINE v	DUP	DIST2	SA:0

This will place the cursor at the line of the editor prompt - for this example, line number 10. The cursor should now be displaying at the beginning of line 10.

To correct the errors found by the assembler, perform the following operations.

- Depress (↓) once - moves cursor down one line.
- Depress (→) 5 times - moves cursor over 5 characters.
- Depress DEL CHAR twice to delete M and comma (,).
- Depress (|←) once - moves cursor to beginning of line.
- Depress (↓) 3 times.
- Depress (→) 3 times.
- Type A to replace letter B with A.
- Depress F1 to terminate the editor operation.

The CRT screen should display:

```
0010  NAM  DIST2
0020  OPT  S
0030  ORG  $2020
0040  LDAA #$AA
>0050  LDAB #10
0060  LDX  #$2000
0070  STAA 0,X
0080  MORE DECB
0090  BNE  MORE
0100  END

>■
F1      F2      F3      F4      F5      F6      F7
CRT     SCROLL  PAGE ^  PAGE v  LINE ^  LINE v  DUP
DIST 2  SA:0
```

To return the system to MDOS,

- Type QUIT(cr) following the Editor prompt (>)

The MDOS prompt (=) should be displayed.

To reassemble Program 2 to check for errors,

Type:

```
=RASM DIST2;AL(cr)
M6800 MACROASSEMBLER 3.xx
COPYRIGHT BY MOTOROLA 1979
```

Reminder! Change "L" to L=#CN if a printer is not part of the system.
---

The output at the printer should be as follows:

```
PAGE 001 DIST2 .SA:0 DIST2

0010 00001          NAM          DIST2
0020 00002          OPT          S
0030 00003A 2020    ORG          $2020
0040 00004A 2020 86 AA A        LDAA          #$AA
0050 00005A 2022 C6 0A A        LDAB          #10
0060 00006A 2024 CE 2000 A       LDX          #$2000
0070 00007A 2027 A7 00 A        STAA         0,X
0080 00008A 2029 5A          MORE DECBB
0090 00009A 202A 26 FD 2029     BNE          MORE
00100 00010        END
TOTAL ERRORS 00000 — 00000
```

MORE 2029

There should be no errors listed. However, there are a few hidden errors:

1. 2000 should be 3000
2. The label MORE should be on line 70 rather than on line 80.
3. Two lines should be inserted: between lines 70 and 80, insert INX; between lines 90 and 100, insert BRA \*

To invoke the CRT editor again,

Type:

```
=E DIST2(cr)
MDOS EDITOR RELEASE 3.xx
COPYRIGHT BY MOTOROLA 1979
```

The CRT screen should display:

```
>0010  NAM  DIST2
0020  OPT  S
0030  ORG  $2020
0040  LDAA #$AA
0050  LDAB #10
0060  LDX  #$2000
0070  STAA 0,X
0080  MORE DECB
0090  BNE  MORE
0100  END
```

EDITING OLD FILE: DIST2 .SA:0 WITH LINE NUMBERS

```
>■
F1      F2      F3      F4      F5      F6      F7
CRT    SCROLL  PAGE ^  PAGE v  LINE ^  LINE v  DUP    DIST2  SA:0
```

To make corrections,

Type:

C 60/2/3/(cr)

Change on line 60 the "2" to a "3".

The CRT should now display the following; note that line 60 has been corrected.

```
0010  NAM  DIST2
0020  OPT  S
0030  ORG  $2020
0040  LDAA #$AA
0050  LDAB #10
>0060  LDX  #$3000
0070  STAA 0,X
0080  MORE DECB
0090  BNE  MORE
0100  END
```

```
>■
F1      F2      F3      F4      F5      F6      F7
CRT    SCROLL  PAGE ^  PAGE v  LINE ^  LINE v  DUP    LTR1   SA:0
```

Type C 20(cr)

This command places the cursor at the first character position of line 20 so that the other corrections and insertions can be done in the page mode.

Depress (↓) 6 times

Depress (DEL CHAR) 4 times

Depress (↑) once

Depress (INS CHAR) once

Type M

Depress (INS CHAR) again

Type 0

Depress (INS CHAR) again

Type R

Depress (INS CHAR) again

Type E

Depress (INS CHAR) again

Depress space bar

Inserts  
—MORE  
on line 70

Depress (|←) once

Depress (↓) once

Depress INS LINE

Type (space)INX to  
insert space INX  
between lines 70 and 80

Depress (|←) once

Depress (↓) 3 times

Depress INS LINE

Type (space)BRA(space)\*

Depress F1



The corrected program should be displayed on the CRT screen.

```
0010  NAM DIST2
0020  OPT S
0030  ORG $2020
0040  LDAA #$AA
0050  LDAB #10
0060  LDX #$3000
0070  MORE STAA 0,X
0071  INX
0080  DECB
0090  BNE MORE
>0091  BRA *
0100  END
```

```
> ■
F1      F2      F3      F4      F5      F6      F7
CRT     SCROLL  PAGE <  PAGE <  LINE <  LINE <  DUP   DIST2  SA:0
```

To resequence the line numbers,

Type RESE(cr)

The resequenced corrected program should now be displayed.

```
0010  NAM DIST2
0020  OPT S
0030  ORG $2020
0040  LDAA #$AA
0050  LDAB #10
0060  LDX #$3000
0070  MORE STAA 0,X
0080  INX
0090  DECB
0100  BNE MORE
>0110  BRA *
0120  END
```

```
> ■
F1      F2      F3      F4      F5      F6      F7
CRT     SCROLL  PAGE ^  PAGE v  LINE ^  LINE v  DUP   DIST2  SA:0
```

To exit from the CRT editor, type QUIT(cr)

This should return the MDOS prompt (=).

### REASSEMBLING PROGRAM 2

Since the program has been assembled previously with the A option, the output file must be deleted.

Type:

```
=DEL DIST2.LO(cr)
DIST2 .LO:0 DELETED
```

To call the assembler,

Type:

```
=RASM DIST2;AL(cr)
M6800 MACROASSEMBLER 3.xx
COPYRIGHT BY MOTOROLA 1979
```

Use L=#CN if a printer is not part of the system.

The output at the printer should be:

```
PAGE 001 DIST2 .SA:0 DIST2

00010 00001          NAM      DIST2
00020 00002          OPT      S
00030 00003A 2020    ORG      $2020
00040 00004A 2020 86 AA A    LDAA  #$AA
00050 00005A 2022 C6 0A A    LDAB  #10
00060 00006A 2024 CE 3000 A   LDX   #$3000
00070 00007A 2027 A7 00 A MORE STAA  0,X
00080 00008A 2029 08          INX
00090 00009A 202A 5A          DECB
00100 00010A 202B 26 FA 2027 BNE   MORE
00110 00011A 202D 20 FE 202D BRA   *
00120 00012          END
TOTAL ERRORS 00000 — 00000
```

MORE 2027

There should be 0 errors.

Program 2 is now available to run on the EXORciser II.

PROGRAM 3

. Editing in CRT Mode

To call the Editor,

Type:

=E DIST3(cr)  
MDOS EDITOR RELEASE 3.XX

>0010 ■

EDITING NEW FILE: DIST3 SA:0 WITH LINE NUMBERS

>

F1	F2	F3	F4	F5	F6	F7		
CRT	SCROLL	PAGE <	PAGE <	LINE <	LINE <	DUP	DIST3	SA:0

Enter Program 3 shown below.

After the program is entered, the display should be as follows:

```
0010  NAM DIST3
0020  OPT S
0030  ORG $2030
0040  LDAB #10
0050  ALSO CMPA 0,X
0060  BNE ERROR
0070  INX
0080  DECB
0090  BNE ALSO
0100  STOP BRA STOP
0110  ERROR LDAB $EE
0120  BRA STOP
>0130  END ■
```

>

F1	F2	F3	F4	F5	F6	F7		
CRT	SCROLL	PAGE ^	PAGE v	LINE ^	LINE v	DUP	DIST3	SA:0

Depress F1

```
>0010  NAM DIST3
0020  OPT S
0030  ORG $2030
0040  LDAB #10
0050  ALSO CMPA 0,X
0060  BNE ERROR
0070  INX
0080  DECB
0090  BNE ALSO
0100  STOP BRA STOP
0110  ERROR LDAB $EE
0120  BRA STOP
>0130  END
```

>■

F1 CRT      F2 SCROLL      F3 PAGE ^      F4 PAGE v      F5 LINE ^      F6 LINE v      F7 DUP      DIST3      SA:0

To terminate the editor,

Type QUIT(cr)

. Assembling Program 3 (1st pass)

To call the assembler,

Type:

```
=RASM DIST3;AL(cr)
M6800 MACROASSEMBLER 3.xx
COPYRIGHT BY MOTOROLA 1979
=
```

Use L=#CN if a printer is not part of the system.

The output at the printer should be:

PAGE 001 DIST3 .SA:0 DIST3

```
00010 00001          NAM          DIST3
00020 00002          OPT          S
00030 00003A 2030    ORG          $2030
00040 00004A 2030 C6 0A A      LDAB          #10
00050 00005A 2032 A1 00 A ALSO CMPA          0,X
00060 00006A 2034 26 06 203C BNE          ERROR
00070 00007A 2036 08          INX
00080 00008A 2037 5A          DECB
00090 00009A 2038 26 F8 2032 BNE          ALSO
00100 00010A 203A 20 FE 203A STOP BRA          STOP
00110 00011A 203C D6 EE A ERROR LDAB          $EE
00120 00012A 203E 20 FA 203A BRA          STOP
00130 00013          END
TOTAL ERRORS 00000 — 00000
```

ALSO 2032 ERROR 203C STOP 203A

Note that there are no errors for this assembly. However, there are several errors that are not apparent. The correct program is listed below.

#### HIDDEN ERRORS

Correct Program 3.

```
0010  NAM DIST3
0020  OPT S
0030  ORG $2030
0035  LDAA #$AA
0040  LDAB #10
0041  LDX #$3000
0050  ALSO CMPA 0,X
0060  BNE ERROR
0070  INX
0080  DECB
0090  BNE ALSO
0100  STOP BRA STOP
0110  ERROR LDAB #$EE
0120  BRA STOP
0130  END
```

The changes: Line 35 - LDAA #\$AA has been inserted  
Line 41 - LDX #\$3000 has been inserted  
On line 110, the \$ has been changed to # \$

Using the CRT editor, make these corrections, then re-assemble the program. An example follows.

. Corrections Program 3

To recall the editor,

Type:

```
=E DIST3(cr)
MDOS EDITOR RELEASE 3.xx
COPYRIGHT BY MOTOROLA 1979
```

The CRT screen will display the following:

```
>0010  NAM DIST3
0020  OPT S
0030  ORG $2030
0040  LDAB #10
0050  ALSO CMPA 0,X
0060  BNE ERROR
0070  INX
0080  DECB
0090  BNE ALSO
0100  STOP BRA STOP
0110  ERROR LDAB $EE
0120  BRA STOP
0130  END
```

EDITING NEW FILE: DIST3 .SA:0 WITH LINE NUMBERS

```
> █
F1      F2      F3      F4      F5      F6      F7
CRT     SCROLL  PAGE ^  PAGE v  LINE v  LINE ^  DUP   DIST3  SA:0
```

To place the cursor at the beginning of the first line,

Type C(cr)

### RE-EDITING PROGRAM 3

To correct Program 3, do the following:

- Depress (↓) 4 times
- Depress (INS LINE) once
- Type (space)LDX(space)#3000
- Depress (|←) once
- Depress (↓) 7 times
- Depress (→) 11 times
- Depress (INS CHAR) once
- Type #
- Depress (|←) once

To terminate page editing mode, depress F1 once.

The CRT screen will display the following:

```
0010  NAM DIST3
0020  OPT S
0030  ORG $2030
0040  LDAB #10
0041  LDX #$3000
0050  ALSO CMPA 0,X
0060  BNE ERROR
0070  INX
0080  DECB
0090  BNE ALSO
0100  STOP BRA STOP
>0110  ERROR LDAB #SEE
0120  BRA STOP
0130  END

>■
F1      F2      F3      F4      F5      F6      F7
CRT     SCROLL  PAGE <  PAGE <  LINE <  LINE <  DUP     DIST3   SA:0
```

Type 35(space)(space)LDAA(space)#\$AA(cr)

Depress F1

The corrected program should be displayed; however, the program needs to be resequenced in case there is a need for further editing later.

```
0010  NAM DIST3
0020  OPT S
0035  ORG $2030
0030  LDAA #$AA
>0040  LDAB #10
0041  LDX #$3000
0050  ALSO CMPA 0,X
0060  BNE ERROR
0070  INX
0080  DECB
0090  BNE ALSO
0100  STOP BRA STOP
0110  ERROR LLAB #$EE
0120  BRA STOP
0130  END

>■
F1      F2      F3      F4      F5      F6      F7
CRT     SCROLL  PAGE ^  PAGE v  LINE ^  LINE v  DUP   DIST2  SA:O
```

Type RESE(cr)

The resequenced corrected program should now be displayed.

```
0010  NAM DIST3
0020  OPT S
0030  ORG $2030
0040  LDAA #$AA
>0050  LDAB #10
0060  LDX #$3000
0070  ALSO CMPA 0,X
0080  BNE ERROR
0090  INX
0100  DECB
0110  BNE ALSO
0120  STOP BRA STOP
0130  ERROR LDAB #$EE
0140  BRA STOP
0150  END

>■
F1      F2      F3      F4      F5      F6      F7
CRT     SCROLL  PAGE <  PAGE <  LINE <  LINE <  DUP   DIST3  SA:O
```

Type QUIT(cr)



. Delete DIST3.LO

Type

=DEL DIST3.LO(cr)

. Assembling the Corrected Program

Type:

=RASM DIST3;AL(cr)  
M6800 MACROASSEMBLER 3.xx  
COPYRIGHT BY MOTOROLA 1979  
=

Use L=#CN if a printer is  
not part of the system.

Printer listing:

```
PAGE 001 DIST3 .SA:0 DIST3

00010 00001          NAM          DIST3
00020 00002          OPT          S
00030 00003A 2030    ORG          $2030
00040 00004A 2030 86 AA A      LDAA          *$AA
00050 00005A 2032 C6 0A A      LDAB          *10
00060 00006A 2034 CE 3000 A     LDX          *$3000
00070 00007A 2037 A1 00 A ALSO CMPA          0,X
00080 00008A 2039 26 06 2041   BNE          ERROR
00090 00009A 203B 08          INX
00100 00010A 203C 5A          DECB
00110 00011A 203D 26 F8 2037   BNE          ALSO
00120 00012A 203F 20 FE 203F STOP BRA          STOP
00130 00013A 2041 C6 EE A ERROR LDAB          *$EE
00140 00014A 2043 20 FA 203F   BRA          STOP
00150 00015          END
TOTAL ERRORS 00000 — 00000
```

All three programs should now be correct and assembled. However, if Program 1 or 2 still has errors, the errors should be corrected and the programs re-assembled at this time.

APPENDIX B

EDITING IN THE SCROLL-MODE AND ASSEMBLY OF PROGRAMS

## OTHER TERMINALS

For terminals other than EXORterm 150, 155, 200, and 220, the Scroll Mode of the CRT Editor must be used. For example, if an EXORterm 100 is to be used, the editing and assembling examples would proceed as follows.

. Call MDOS

Type:

```
*E MDOS(cr)
MDOS 3.xx
=
```

. Call the Editor

Type:

```
=E DIST1;S(cr)
MDOS EDITOR RELEASE 3.xx
COPYRIGHT BY MOTOROLA 1979
```

The following should be displayed on the screen:

```
EDITING NEW FILE : DIST .SA:0 WITH LINE NUMBERS
0010
```

## EDITING IN SCROLL MODE

### . Program 1

Type in the following program exactly as shown.

NOTE: If a printer is not part of the system, change line 20 to OPT S,NOP.

EDITING NEW FILE : DIST1.SA:0 WITH LINE NUMBERS

```
0010  NAM DIST1 (CR)
0020  OPT S (CR)
0030  ORG $2000 (CR)
0040  LDX #$3000 (CR)
0050  CLRA (CR)
0060  LDAB #10 (CR)
0070  AGAIN STAA 0,X (CR)
0080  INX (CR)
0090  DECB (CR)
0100  BNE AGAIN (CR)
0110  BRA *(CR)
0120  END (CR)
0130  (CR)
```

Type QUIT(cr) to terminate the editing process and save the file on the diskette (DIST1.SA).

The MDOS prompt (=) should now be displayed.

ASSEMBLING PROGRAM 1

. Call the Assembler

Type:

=RASM DIST1;AL  
M6800 MACROASSEMBLER 3.xx  
COPYRIGHT BY MOTOROLA 1979

Reminder! Change "L" to L=#CN if a printer is not part of the system.
---

The program should appear at the printer as follows:

```
PAGE 001 DIST1 .SA:0 DIST1
00010 00001          NAM      DIST1
00020 00002          OPT      $
00030 00003A 2000    ORG      $2000
00040 00004A 2000 CE 3000      A      LDX      #$3000
00050 00005A 2003 4F          CLRA
00060 00006A 2004 C6 0A      A      LDAB     #10
00070 00007A 2006 A7 00      A AGAIN STAA    0,X
00080 00008A 2008 08          INX
00090 00009A 2009 5A          DECB
00100 00010A 200A 26 FA 2006    BNE     AGAIN
00110 00011A 200C 20 FE 200C    END     BRA     *
00120 00012
TOTAL ERRORS 00000-00000

AGAIN 2006
```

There should be 0 errors. If there are any errors, they were introduced by the user and will be corrected later.

The MDOS prompt (=) should be displayed on the screen.

EDITING PROGRAM 2

. Program 2

Type:

=E DIST2;S(cr)

MDOS EDITOR RELEASE 3.xx  
COPYRIGHT BY MOTOROLA 1979

Enter the program exactly as shown below; if not using a printer, change line 20 to OPT M,S,NOP.

EDITING NEW FILE : DIST2 .SA:0 WITH LINE NUMBERS

```
0010  NAM DIST2 (CR)
0020  OPT M,S (CR)
0030  ORG $2020 (CR)
0040  LDAA #$AA (CR)
0050  LDBB #10 (CR)
0060  LDX #$2000 (CR)
0070  STAA 0,X (CR)
0080  MORE DECB (CR)
0090  BNE MORE (CR)
0100  END(CR)
0110  (CR)
>
```

Type QUIT(cr) to save the file and return system to MDOS.

The MDOS prompt (=) should now be displayed.

ASSEMBLING PROGRAM 2

Type:

=RASM DIST2;AL(cr)  
M6800 MACROASSEMBLER 3.xx  
COPYRIGHT BY MOTOROLA 1979

Use L=#CN if a printer is not part of the system.
--

The following should appear at the printer:

PAGE 001 DIST2 .SA:0 DIST2

0010	00001				NAM	DIST2
0020	00002				OPT	M,S
0030	00003A	2020			ORG	\$2020
***ERROR 235--00000						
	00004A	2020 86	AA	A 0040	LDAA	#\$AA
***ERROR 207--00004						
	00005A	2022 00	0000	A 0050	LDBB	#10
0060	00006A	2025 CE	2000	A	LDX	#\$2000
0070	00007A	2028 A7	00	A	STAA	0,X
0080	00008A	202A 5A		MORE	DECB	
0090	00009A	202B 26	FD 202A		BNE	MORE
0100	00010				END	

TOTAL ERRORS 00002--00005

MORE 202A

- The MDOS prompt (=) should appear on the CRT screen.
- There should be only 2 errors; any others were user generated:
  - . Error 235 because the "M" should not be in the OPT statement.
  - . Error 207 because the first "B" in LDBB should be an "A".
- To correct these errors, the editor will have to be invoked again.

RE-EDITING PROGRAM 2

Type:

=E DIST2;S(cr)  
MDOS EDITOR RELEASE 3.xx  
COPYRIGHT BY MOTOROLA 1979

The screen should display:

```
EDITING OLD FILE : DIST2   .SA:0 WITH LINE NUMBERS  
>
```

Type L(cr) to list the file being edited.

The old file should now be displayed.  
The > is the editor prompt. The editor is awaiting a new command.  
The edit commands are listed in Table 1-1.

```
EDITING OLD FILE : DIST2   .SA:0 WITH LINE NUMBERS
```

```
0010   NAM DIST2  
0020   OPT M,S  
0030   ORG $2020  
0040   LDAA #$AA  
0050   LDBB #10  
0060   LDX #$2000  
0070   STAA 0,X  
0080  MORE DECB  
0090   BNE MORE  
0100   END
```

```
>
```



Type:

```
>C 20/M,/(cr)
0020 OPT S
```

C is the change command;  
i.e., change on line 20 - remove "M,"

Type:

```
>C 50/B/A/(cr)
0050 LDAB #10
```

Change on line 50 - "B" to "A"

Type L(cr)

```
>L (CR)
0010 NAM DIST2
0020 OPT S
0030 ORG $2020
0040 LDAA #$AA
0050 LDAB #10
0060 LDX #$2000
0070 STAA 0,X
0080 MORE DECB
0090 BNE MORE
0100 END
>
```

The known errors are corrected. However, the assembler cannot detect missing statements. The listing below shows the correct program.

```
0010 NAM DIST2
0020 OPT S
0030 ORG $2020
0040 LDAA #$AA
0050 LDAB #10
0060 LDX #$3000
0070 MORE STAA 0,X
0080 INX
0090 DECB
0100 BNE MORE
0110 BRA *
0120 END
```

Correct Program 2

Compare the above program with the program being displayed.  
Notice on line 60 that 2000 should be 3000.

Change on line 60: "2" to "3"  
(Note that the CRT "SCROLLS" after each correction.)

Type:

>C 60/2/3/(cr)  
0060 LDX #\$3000

Change on line 80: delete "MORE"

Type:

>C 80/MORE// (cr)  
0080 DECB

Change on line 70: insert "MORE" in space preceding "STAA 0,X"

Type:

>C 70//MORE/ (cr)  
0070 MORE STAA 0,X

"INX" should be inserted between lines 70 and 80

Type:

>75(space)(space)INX(cr)

"BRA\*" should be inserted between lines 90 and 100

Type:

>95(space)(space)BRA(space)\*(cr)

The program should be resequenced to make the line numbers multiples of ten.

Type:

>RESE(cr)  
0110 BRA\*

Type L(cr)

The program listed on the screen should now be the same as the correct version given on page B-8.

Type QUIT(cr) to terminate the edit operation.

The MDOS prompt (=) should now be displayed.

RE-ASSEMBLING PROGRAM 2

- Since program 2 was assembled once before, the output generated by the assembler must be deleted from the diskette. To do this,

Type:

```
=DEL DIST2.LO(cr)
DIST2.LO:0 DELETED
```

- Call the Assembler

Type:

```
=RASM DIST2;AL(cr)
M6800 MACROASSEMBLER 3.xx
COPYRIGHT BY MOTOROLA 1979
```

Use L=#CN if a printer is not part of the system.

The listing at the printer should be:

PAGE 001 DIST2 .SA:0 DIST2

00010	00001					NAM	DIST2
00020	00002					OPT	S
00030	00003A	2020				ORG	\$2020
00040	00004A	2020 86	AA	A		LDAA	#\$AA
00050	00005A	2022 C6	0A	A		LDAB	#10
00060	00006A	2024 CE	3000	A		LDX	#\$3000
00070	00007A	2027 A7	00	A MORE		STAA	0,X
00080	00008A	2029 08				INX	
00090	00009A	202A 5A				DECB	
00100	00010A	202B 26	FA	2027		BNE	MORE
00110	00011A	202D 20	FE	202D		BRA	#
00120	00012					END	

TOTAL ERRORS 00000--00000

MORE 2027

- There should be no errors.
- The MDOS prompt (=) should be displayed.
- Verify that the correct program was assembled.

PROGRAM 3

Edit and assemble the following program:

```
0010  NAM DIST3
0020  OPT S
0030  ORG $2030
0040  LDAB #10
0050  ALSO CMPA 0,X
0060  BNE ERROR
0070  INX
0080  DECB
0090  BNE ALSO
0100  STOP BRA STOP
0110  ERROR LDAB $EE
0120  BRA STOP
0130  END
```

. Hints: using the previous example as a guide,

- Call the CRT editor
- Call the macro assembler
- If errors, re-edit and re-assemble until zero errors
- If no errors, see next section

. Hidden Errors

Assembly listing for program 3 should be:

PAGE 001 DIST3 .SA:0 DIST3

```
0010 00001          NAM      DIST3
0020 00002          OPT      S
0030 00003A 2030    ORG      $2030
0040 00004A 2030 C6 0A      A      LDAB     #10
0050 00005A 2032 A1 00      A ALSO  CMPA    0,X
0060 00006A 2034 26 06 203C BNE     ERROR
0070 00007A 2036 08          INX
0080 00008A 2037 5A          DECB
0090 00009A 2038 26 F8 2032 BNE     ALSO
0100 00010A 203A 20 FE 203A STOP  BRA     STOP
0110 00011A 203CD6 EE      A ERROR LDAB    $EE
0120 00012A 203E 20 FA 203A BRA     STOP
0130 00013          END
```

TOTAL ERRORS 00000--00000

ALSO 2032 ERROR 203C STOP 203A

However, the correct program 3 should be:

```
0010  NAM DIST3
0020  OPT S
0030  ORG $2030
0035  LDAA #$AA
0040  LDAB #10
0045  LDX #$3000
0050  ALSO CMPA 0,X
0060  BNE ERROR
0070  INX
0080  DECB
0090  BNE ALSO
0100  STOP BRA STOP
0110  ERROR LDAB #SEE
0120  BRA STOP
0130  END
```

Notice:

- . Line 45 has been added
- . On line 110, "#" has been added
- . Add a line 35: 35 LDAA #\$AA

Using the CRT editor, make these corrections and re-assemble program 3.

An example of correcting program 3 using the scroll mode of the CRT editor.

Call the Editor

```
=E DIST3;S(cr)
MDOS EDITOR RELEASE 3.xx
COPYRIGHT BY MOTOROLA 1979
```

```
EDITING OLD FILE : DIST3 .SA:0 WITH LINE NUMBERS
```

```
>L (CR)
```

```
0010  NAM DIST3
0020  OPT S
0030  ORG $2030
0040  LDAB #10
0050  ALSO CMPA 0,X
0060  BNE ERROR
0070  INX
0080  DECB
0090  BNE ALSO
0100  STOP BRA STOP
0110  ERROR LDAB #EE
0120  BRA STOP
0130  END
```

```
>35 LDAA #AA (CR)
```

```
>45 LDX #3000 (CR)
```

```
>C 110$/#$/ (CR)
```

```
0100 ERROR LDAB #EE
```

```
>L (CR)
```

```
0010  NAM DIST3
0020  OPT S
0030  ORG $2030
0035  LDAA #AA
0040  LDAB #10
0045  LDX #3000
0050  ALSO CMPA 0,X
0060  BNE ERROR
0070  INX
0080  DECB
0090  BNE ALSO
0100  STOP BRA STOP
0110  ERROR LDAB #EE
0120  BRA STOP
0130  END
```

```
>RESE (CR)
```

```
0130 ERROR LDAB #EE
```

```

>L (CR)
0010  NAM DIST3
0020  OPT S
0030  ORG $2030
0040  LDAA #$AA
0050  LDAB #10
0060  LDX #$3000
0070  ALSO CMPA 0,X          (CRT)
0080  BNE ERROR
0090  INX
0100  DECB
0110  BNE ALSO
0120  STOP BRA STOP
0130  ERROR LDAB #$EE
0140  BRA STOP
0150  END

<u>QUIT (CR)</u>

```

Assembling the corrected program:

```

=RASM DIST3;AL(cr)
MDOS MACROASSEMBLER 3.xx
COPYRIGHT BY MOTOROLA 1979
=

```

Use L=#CN if a printer is not part of the system.

Printer Listing:

```

PAGE 001 DIST3      .SA:0 DIST3

00010 00001          NAM      DIST3
00020 00002          OPT      S
00030 00003A 2030    ORG      $2030
00040 00004A 2030 86 AA      A    LDAA     #$AA
00050 00005A 2032 C6 0A      A    LDAB     #10
00060 00006A 2034 CE 3000    A    LDX      #$3000
00070 00007A 2037 A1 00      A    ALSO    CMPA    0,X
00080 00008A 2039 26 06     2041 BNE     ERROR
00090 00009A 203B 08          INX
00100 00010A 203C 5A          DECB
00110 00011A 203D 26 F8     2037 BNE     ALSO
00120 00012A 203F 20 FE     203F STOP    BRA     STOP
00130 00013A 2041 06 EE      A    ERROR    LDAB     #$EE
00140 00014A 2043 20 FA     203F BRA     STOP
00150 00015          END

TOTAL ERRORS 00000--00000

ALSO 2037 ERROR 2041 STOP 203F

```

All three programs should now be corrected and assembled. However, if Program 1 or 2 had errors, these errors should be corrected and the programs re-assembled at this time.

# SUGGESTION/PROBLEM REPORT

Motorola welcomes your comments on its products and publications. Please use this form.

To: Motorola Microsystems  
P.O. Box 20912  
Attention: Publications Manager  
Mail Drop M374  
Phoenix, Az. 85036

Comments

Product: \_\_\_\_\_

Manual: \_\_\_\_\_

*Please Print*

\_\_\_\_\_  
Name

\_\_\_\_\_  
Title

\_\_\_\_\_  
Company

\_\_\_\_\_  
Division

\_\_\_\_\_  
Street

\_\_\_\_\_  
Mail Drop

\_\_\_\_\_  
Phone Number

\_\_\_\_\_  
City

\_\_\_\_\_  
State

\_\_\_\_\_  
Zip

HARDWARE SUPPORT: (800) 528-1908  
SOFTWARE SUPPORT: (602) 831-4108



