PERCOM

MPX/9™

USERS MANUAL

PERCOM DATA COMPANY
211 N. KIRBY
GARLAND, TEXAS 75042

Tech Help
214-494-0428

the MPX/9(tm)
6809 Disk Operating System

USERS MANUAL

Preliminary

IMPORTANT NOTICE

All material in this manual is copyrighted by PERCOM DATA CO. INC. No portion of it may be copied or reproduced in any manner without the written permission of PERCOM DATA CO. INC.

Although the information contained in this publication has been thoroughly checked for accuracy and reliability, PERCOM DATA CO. INC. shall have no liability or responsibility to customer or any other person or entity with respect to any liability, loss or damage caused or alleged to be caused directly or indirectly by products or programs sold by PERCOM DATA CO. INC., including but not limited to any interruption of service, loss of business or anticipatory profits or consequential damages resulting from the use or operation of such products or programs. Furthermore, PERCOM DATA CO. INC. does not represent the described equipment or programs as suitable for any purpose and does not assume any liability arising out of the application or use of any product, circuit or program described herein.

PERCOM DATA CO. INC. reserves the right to make changes to any products or specifications described herein without notice.

TRADEMARKS USED IN THIS MANUAL ARE INDICATED AS FOLLOWS:

(tm) = trademark of Percom Data Company, Inc.

~ = trademark of Technical Systems Consultants, Inc.

TABLE OF CONTENTS

PERCOM DATA CO. INC.
211 N. Kirby
Garland, TX 75042
(214) 272-3421

NOTICE

All COMPUTER PROGRAMS sold or distributed by PERCOM DATA CO. INC. are sold or distributed on an AS-IS basis WITHOUT WARRANTY.

PERCOM DATA CO. INC. shall have no LIABILITY or responsibility to customers, or any other person or entity with respect to any LIABILITY, LOSS, OR DAMAGE caused or alleged to be caused directly or indirectly by equipment or computer programs sold by PERCOM DATA CO. INC. including but not limited to any interruption of service, loss of business or anticipatory profits or consequential damages resulting from the use or operation of such equipment or computer programs.

Good data processing procedure dictates that the user test the program, run and test sample sets of data, and run the system in parallel with the system previously in use for a period of time adequate to insure that results of operation of the computer or program are satisfactory.

I INTRODUCTION

The MPX/9(tm) is a 6809 disk operating system (DOS) for the Percom series of LFD disk storage systems.

MPX/9(tm) consists of a 1-Kbyte ROM with low-level drivers accessible through device control blocks (DCBs), and a minidiskette containing the command and utility processor plus a "boot" for loading the minidiskette code into system RAM.

The MPX/9(tm) features position-independent code. This allows the user to install the MPX/9(tm) ROM in any of the three sockets provided on the LFD disk system controller card. Moreover, it means the minidiskette code may be loaded into any unprotected RAM space.

Besides position-independent code, the MPX/9(tm) also features mnemonic error reporting codes; file manipulation by name instead of by drive and sector numbers as is done under MINDOS/MPX, the Percom 6800 two-ROM DOS; and, the accommodation of disk-resident transient commands which permit MPX/9(tm) to be extended indefinitely.

For Percom LFD disk systems presently equipped with MINIDOS(tm)/MPX(tm), the addition of MPX/9(tm) will extend the disk system for operation with either a 6800 or 6809 computer, without contention between systems.

II SYSTEM REQUIREMENTS

2.1 DISK SYSTEM

The MPX/9(tm) operates the Percom LFD series of disk systems. The MPX/9(tm) ROM may be installed in any of the three ROM sockets provided on the LFD controller card. However, if the MINIDOS/MPX system is presently installed, the MPX/9(tm) ROM must be installed in socket #3 for harmonious operation with either the 6800 MINIDOS/MPX system or the 6809 MPX/9(tm) system.

2.2 COMPUTER SYSTEM SOFTWARE

The MPX/9(tm) is designed to operate with PSYMON(tm), the system monitor for the Percom SBC/9(tm) Single-Board Computer/MPU card.

*** CAUTION ***

MPX/9(tm) works only with PSYMON(tm) version 1.10 or later. A version printed in '68' Micro Journal magazine will not work with MPX/9(tm).

2.3 SYSTEM MEMORY

The MPX/9(tm) ROM code resides in 1-Kbyte of memory space, and is addressed at C000, C400 or C800 (hex) depending on whether the ROM IC is installed in socket #1, #2 or #3 on the LFD controller card. The program initializes RAM and loads the "boot" on the minidiskette into 0100 (hex).

The minidiskette code self-loads into 4-Kbytes, beginning at the first unprotected 4-Kbyte boundary below the disk controller. The LFD controller (MINIDOS/MPX) is normally mapped at C000 (hex).

*** CAUTION ***

The MPX/9(tm) minidiskette program requires a full 4 Kbytes of RAM. An error will result if the starting point found by the "boot" is less than four Kbytes.

2.4 OPERATION WITH FLEX~

For operation with TSC's FLEX~ for the 6809, the LFD controller must be remapped from C000 (hex) to E000 (hex). The TSC 6809 FLEX~ requires 8-Kbytes of RAM located at C000 - DFFF (hex).

The LFD controller may be modified for operation at E000 (hex) as follows:

1. Cut the circuit trace between pin 3 of B10 and pin 40 of the edge connector.
2. Connect a jumper between pin 40 of the edge connector and pin 6 of B1.
3. Connect a jumper between pin 7 and pin 3 of B10.

III COMMANDS

3.1 COMMAND DESCRIPTIONS and FORMATS

Commands are entered in response to the system prompt: MPX ?. This prompt was chosen to avoid confusion with the PSYMON(tm) prompt, CMD ?.

The MPX/9(tm) memory-resident commands are defined by single characters -- for example, S for SAVE. Disk-resident transient commands are defined by multiple characters -- for example, COPY for the file copy utility.

MPX/9(tm) supports the following memory-resident commands:

COMMAND	ENTRY FORMAT*
S(ave)	S <FNAME> <BEGA> <ENDA> [<EXEC>]
L(oad)	L <FNAME> (Address)
D(elete)	D <FNAME>
F(iles)	F <N>
I(nitial.)	I <N> [<CMD>]
R(ename)	R <OLDNAME> <NEWNAME>
G(o)	G <EXEC> [<parameters>]
M(onitor)	M --

* Blanks are used as delimiters to separate the entries of a line. Entries enclosed in brackets, [ ], are optional. The angle brackets, < and >, are not included with an entry.

The command line entries of the above table are defined as follows:

COMMANDS - Commands are the literals S, L, D, etc. If a multi-letter command is entered, the system will search for a disk file with the name of the command.

<FNAME> - File name. File names may be entered in several forms.

Examples:

REDACTOR	(name only)
3/REDACTOR.TX	(drive no.)(name)(suffix)
REDACTOR.TX,40	(name)(suffix)(size)

For the general case, <FNAME> has the form:

N/NAME.SF,SIZE

where

a. N/ specifies the drive number, N = 1, 2, 3 or 4. If a drive number is not entered, the system defaults to drive #1.

*** NOTE ***

In MPX/9(tm), the drive number is only specified as part of the name. It is not specified as part of the command as for MINIDOS/MPX. If a command is preceded by a drive number, the system will assume the command is disk resident.

b. NAME is the name of the file, and may be up to eight alphanumeric characters.

c. SF is a one- or two-character suffix that further defines the file. If a suffix is not entered, the defaults are CM for file names entered as commands, and MI (Memory Image) for file names referenced by LOAD and SAVE commands. Suffixes are defined below.

d. SIZE is a decimal value for the number of blocks required to store the file. Size must be entered only for new files. The size default is eight blocks if a size is not provided. In some cases -- for example in using the SAVE command -- the size is implicitly computed.

<BEGA> - Beginning address. Enter the beginning address as a hexadecimal value.

<ENDA> - Ending address. Enter as a hexadecimal value.

<EXEC> - Optional program start address for SAVE command; mandatory entry for GO command. Enter as a hexadecimal value.
If an execution address is not supplied when the program is S(aved), it will not execute automatically as a command file, and an MPX/9(tm) GO command must be used to cause the program to execute.

(Address) - Optional entry for LOAD command. May be used to load a file to an address other than the one it was S(aved) from. Enter address as a hexadecimal value.

<N> - Drive number. <N> = 1, 2, 3 or 4.

<CMD> - Optional entry for Initialize command. May be any legal MPX/9(tm) command. See the discussion of the Initialize command below for details.

<OLDNAME> <NEWNAME> - File names. Refer to discussion of <FNAME> above.

<parameters> - Any information that is to pass to the program being executed. The source file name and the destination file name for a COPY operation are examples of parameters.

3.2 COMMAND DESCRIPTIONS and PROCEDURES

3.2.1 Preliminary

The MPX/9(tm) disk operating system communicates with the data terminal by calling I/O subroutines in PSYMON(tm).

1. Insert the MPX/9(tm) minidiskette in drive #1.

2. Using the PSYMON(tm) GO command, enter MPX/9(tm) at C000, C400 or C800 (hex), depending on whether the MPX/9(tm) ROM is in socket #1, #2 or #3, respectively, of the LFD controller card.

At this point, the MPX/9(tm) ROM loads the "boot" from the minidiskette into address 0100 (hex). If an error occurs during the booting, control returns to PSYMON(tm), and a numeric error code will be displayed. Error codes are defined in paragraph 3.3.

MPX/9(tm) commands are entered into a 64-character buffer. No action is taken until the receipt of a RETURN code. This permits errors to be edited during entry.

*** NOTE ***

1. Control H will backspace in the command line.
2. Control X will cancel the line and reprompt.
3. A RETURN code terminates the command input.

3.2.2 MPX/9(tm) Memory Resident Commands

3.2.2.1 I - INITIALIZE

The INITIALIZE command, I, initializes a disk. A disk directory must be initialized before the disk can be used by MPX/9(tm).

*** CAUTION ***

Files on a disk are lost when the disk is initialized. Consequently, the system will prompt:

```
ENTER Y TO INIT. DISK IN DRIVE (N)
```

when an I(nitialize) command is entered. Responding by entering any character other than Y will result in error code OK being displayed, and the initilization attempt will be aborted. If the letter Y is entered, the disk will be initialized.

Format:

```
I [<N>] [<CMD>], where <N> = 1, 2, 3 or 4.
```

Both <N> and <CMD> are optional, as indicated by the brackets. If a drive number is not entered, the system defaults to drive #1. If a command is entered, it will be executed automatically when booting from the disk.

Examples:

```
I
I 3
I 2 M
```

3.2.2.2 S - SAVE

The SAVE command, S, is used to save a program in memory onto a disk by either creating a new file or overwriting an existing file by the same name.

Example:

```
S FNAME 100 1BFF 200
         |    |   |
         |    |   |_________Start Address (optional)
         |    |_____________________________Ending address
         |_____________________________Beginning address
```

Note: Leading zeroes are not required for addresses.

3.2.2.3 L - LOAD Command

The LOAD command, L, causes a program to be loaded into memory but does not initiate execution. This feature is useful for making changes in a program before it is executed.

To load and execute, simply enter the program name without the L command. The file will be located, loaded into memory and executed -- provided a program start address was supplied when the program was saved to disk.

The command line may include an optional address to load a file to an address other than the one it was saved from.

Examples:

L BASIC	A Program named BASIC.MI is loaded from drive #1 but not executed. After loading, system control returns to MPX/9(tm).
3/BASIC	A Program named BASIC.CM is loaded from drive #3 and executed.
L 3/BASIC $4000	A program named BASIC.MI is loaded beginning at address 4000(hex) from a disk in drive #3.

3.2.2.4 F - FILES Command (Lists a File Directory)

The FILES command, F, causes the disk directory to be displayed on the data terminal. The following information is reported for each file:

File Name	(FILE NAME)
First Block Used	(START) (decimal block no.)
Last Block Used	(END) (decimal block no.)
Program Start Address	(EXT) (hexadecimal)

The system displays information about 14 files then displays the prompt 'CR TO CONTINUE' and waits until a carriage return is entered. Entering a RETURN will cause the next 14 (or less) files to be displayed. The remaining files, if any, may be displayed by entering RETURN again.

If the program start address is given as FFFF, a program start address was not supplied when the file was saved, and automatic execution is inhibited.

Example:

```
F 3
```

3.2.2.5 R - RENAME

The RENAME command, R, is used to change a file name.

Examples:

```
R OLDNAME NEWNAME
```

```
R 2/OLDNAME NEWNAME
```

The first command changes a file name on drive #1. The second command changes a file name on drive #2.

3.2.2.6 D - DELETE

The DELETE command, D, is used to delete files from a disk. The space formerly taken by a deleted file is available for storing other files.

Examples:

```
D OBSOLETE       Deletes file OBSOLETE on drive #1
D 4/OUTDATED     Deletes file OUTDATED on drive #4
```

3.2.2.7 M - MONITOR

The MONITOR command, M, returns control to PSYMON(tm), the system monitor. The literal M is the only entry on the command line.

3.2.2.8 G - GO

The GO command, G, is used to go to a specified address and begin execution.

Examples:

```
G 1000
G 1000 NAMEONE NAMETWO
```

The file names, NAMEONE and NAMETWO, are optional parameters, which are discussed in paragraph 3.1.

3.2.3 Disk-Resident (Transient) Commands

The MPX/9(tm) disk-resident commands are entered in response to the prompt, MPX ?, as discussed above. Disk-resident commands are described in an appendix to this manual.

*** NOTE ***

The MPX/9(tm) utilities default execution address is 0100(hex). Since the utilities are written in position-independent code, they may be loaded into an alternate address using the MPX/9(tm) memory-resident LOAD command. The GO command is used to initiate execution:

```
G <EXEC> <parameters>
```

3.3 ERROR CODES

The MPX/9(tm) reports errors as two-character mnemonic codes -- except during "booting" when errors are reported by numeric codes.

3.3.1 Mnemonic Error Codes

OK	- No error
FN	- Illegal FuNction attempted
ID	- Illegal Drive number
IB	- Illegal Block number
DM	- Disk Missing
NB	- Null (empty) Block read
SK	- SeeK error
RD	- Disk ReaD error
VF	- Disk VeriFy error
WP	- Write-Protected disk
NF	- File Not found
DF	- Disk Full, or no space for file
IF	- Invalid File specification
NC	- File Not Closed
UF	- Unopened File
IA	- Illegal Access to file
EF	- End of File
SN	- SyNtax error in command

3.3.2 Numeric Error Codes

0	- No I/O error, but no MPX/9(tm) on the program disk.
1	- Illegal function code
2	- Illegal drive number
3	- Illegal block number
4	- Disk missing or door open
5	- Empty block read (no data)
6	- Seek error
7	- Read error
8	- Verify error (after write)
9	- Write-protected disk

3.4 STANDARD FILE SUFFIXES

The standard MPX/9(tm) two-character suffixes for file names are:

CM	- Command file
MI	- Memory image file
SY	- System file
TX	- Text file
RS	- Reserved space

HX - Hexadecimal (S1 - S9) file

The MPX/9(tm) defaults to the CM suffix for file names entered as commands, and to the MI suffix for file names referenced by the LOAD and SAVE commands.

COPY UTILITY

This MPX/9(tm) utility may be used to copy files from one disk to another disk. Files of any type may be copied. All disk writes are verified.

Procedure:

The COPY utility is loaded from the MPX/9(tm) system disk. There are three command line formats. Blanks are used as delimiters between the entries on a command line. Errors are reported in MPX/9(tm) codes.

FORMAT #1:

```
COPY <SOURCE> <DESTINATION>
```

The first file name is the source file and the second file name is the destination. (File names may include drive numbers as discussed in Section III.) The copying is performed from source to destination.

Examples:

```
COPY   SRCENAME DESTNAME      (Utility in drive #1)
4/COPY SRCENAME DESTNAME     (Utility in drive  #4)
```

FORMAT #2:

```
COPY <FNAME> <N>
```

The named file is copied to a file of the same name on the disk in drive N.

Examples:

```
COPY NAMETEST 2
COPY COPY  3
```

FORMAT #3:

```
COPY <N1> <N2>   where N1, N2 = 1, 2, 3 or 4.
```

All files on the disk in drive N1 are copied to a disk in drive N2.

Example:

```
COPY 3 2
```

-more-

After a copy operation is performed, the system will prompt: 'COPY' and wait for another command. To return to MPX/9(tm), strike the carriage return.

The COPY utility will also recognize:

```
? N    where N = 1, 2, 3 or 4
```

to display the files on a drive.

REMAP UTILITY

This utility will remap a 6800 MINIDOS/MPX diskette for compatibility with the MPX/9(tm) operating system. REMAP constructs an MPX/9(tm) directory from a 6800 MINIDOS/MPX directory. After the directory has been rebuilt, each file is read block by block and the block headers are restructured to conform to MPX/9(tm) conventions.

*** CAUTION ***

Since the conversion is done in place, a backup should be made prior to performing a remap. The 6800 MPX COPY utility may be used to make a backup of the 6800 disk.

*** NOTE ***

The REMAP utility will only process 'named' file entries in a 6800 directory. If MINIDOS/MPX DSSS type files -- disk/sector number files -- must be converted, use the 6800 MPX COPY utility to create a name directory entry. The 6800 MPX command to do this is 'COPY DSSS NAME', where DSSS is the four digit drive and Sector number and NAME is the name of the file.

Only 30 files may be converted due to the addition of a directory entry for the disk directory itself in MPX/9(tm). If an attempt is made to remap a disk containing 31 or more files, a warning message is issued.

A test is made prior to a remapping to ensure that the disk is not already in MPX/9(tm) format, to prevent accidental destruction of MPX/9(tm) files.

The names of remapped files are given the suffix 'CV' (converted) so they may be recognized as remapped files.

-more-

During the remapping, all writes are done with read-after verification.

Procedure

The REMAP utility is loaded from the MPX/9(tm) system disk. The command format is as follows:

```
        REMAP <N>    where <N> = 1, 2, 3 or 4.
```

Errors are reported in MPX/9(tm) two-character codes. Errors which occur during directory I/O are considered fatal. An error which occurs during a file conversion will cause a display of 'CONTINUE?' after the error is reported. For any response other than 'Y', the remapping will be aborted, while a 'Y' response will cause remapping to resume with the next file.

end - Appendix A1

HOW TO ORDER PARTS AND OPTIONAL ITEMS

HOW TO ORDER: Order by mail...we're as near as your mailbox... or order by phone.

TOLL-FREE PHONE ORDERS: To save you money and insure prompt service, we've installed a toll-free number: 1-800-527-1592 FOR PLACING ORDERS ONLY. In Texas, and for Customer Service, dial (214) 272-3421. We cannot transfer calls received on our toll-free number to other departments -- please help us serve you better by dialing the correct number.

PROMPT SERVICE: We ship the cheapest, fastest way. We use UPS up to 50 lbs. per item, 100 lbs. per shipment. We use truck-freight for large or heavy shipments. Transportation charges collected on delivery.

COD ORDERS: COD orders are accepted where possible.

OPEN ACCOUNT TERMS: Net 10 days to rated firms.

TEXAS SALES TAX: Texas law requires that we collect 5% sales tax on all shipments in Texas.

MINIMIUM ORDERS: We will add a handling charge of $2.00 to all orders totalling less than $15.00.

RETURNS: Write or call for return authorization before returning any merchandise. RETURNS WITHOUT AUTHORIZATION WILL BE REFUSED.

DAMAGED SHIPMENTS: Have carrier note if received in damaged condition, then file claim. About concealed damage: contact carrier for inspection, then file claim. Save the shipping carton.

PERCOM STANDARDS For MPX-9 (tm) Compatible Software

UTILITY PROGRAMS

1) All utility programs must be written in Position Independent Code (PIC).
2) All utility programs must be 'origined' to location 0.
3) The starting point of all utility programs must be location 0 which may be a branch or program counter relative jump to the actual starting location.
4) All utility programs must exit via an RTS regardless of the error condition.
   a) 'Z' flag in Condition Code (CC) register is set if there was NO error.
   b) 'Z' flag in CC register is clear if there was an error.
5) If the utility exits after an error occurs, generation of an error report should not be attempted by the utility. The error code should be loaded into ACC 'B' and exit should be via a 'TSTB,RTS' sequence.

DEVICE DRIVERS

1) All device drivers must be written in Position Independent Code (PIC).
2) All device drivers must be 'origined' to location 0.
3) The starting point of all device drivers must be location 0 which may be a branch or program counter relative jump to the actual starting location.
4) The device driver will be composed of two (2) parts.
   a) Part 1 will be the first section. It will be executed when the device driver is 'run'. It must set up the device DCB and initialize the device.
   b) Part 2 will consist of the drivers and the DCB. It is the only section actually used by the system after initialization. The driver should follow all the rules given in the PSYMON (tm) manual. Provision may be made for deletion of the device driver when a EOT (4) code is written to the device if you so desire.
5) ONLY THE 'A' REGISTER AND THE 'CC' REGISTER MAY BE ALTERED BY PART 2 OF THE DEVICE DRIVER. PART 1 MAY USE ALL REGISTERS.
6) Rule 4 of 'UTILILITY PROGRAMS' applies as regards errors. Any error must be returned in DCBERR as well.
7) WRITE ONLY drivers, such as printer drivers, must return EOT (4) immediately if called for READ.
8) READ ONLY drivers should simply return if called for WRITE.

* NOTE: The classification of (7) and (8) as error conditions or as allowable conditions is left to the programmer.

CERTIFY - CERTIFY THE USABILITY OF A DISKETTE
AN MPX-9 UTILITY PROGRAM

This utility will allow you to check a diskette prior to use to be certain that it has no bad spots on it. It will start at sector 0 and format all sectors with 256 bytes of $04. It will the verify that all sectors are readable and still contain the same information. Load @ 1000

```
MPX? [N/]CERTIFY <drive> [/S]
```

This command will fetch the CERTIFY utility from drive #N, and certify the disk in drive # <drive>. If desired, you may specify an ending sector other than the default value of 399. The value, S, must be preceeded by a '/' mark.

Disk certification may be performed with a single drive system as the utility will ask if you are ready prior to starting. Simply remove your utility diskette, insert the diskette that you wish to certify, then answer with a 'Y'.

You will be appraised of all bad sectors.

The default value of 399 as the last sector is found at the beginning location of the utility +1. It occupies 2 bytes, and is given in hexadecimal. It may be altered and the utility resaved if desired.

RELATIVE ADDRESS	
0000	START of code.
0001	16 bit default value for last sector currently 399 (hex 018F).
007F	Displays sector numbers on formatting operation.
0086	NOP ($12) these bytes if you are not using a CRT.
00AD	Displays sector numbers on verification operation.
00B4	NOP ($12) these bytes if you are not using a CRT.
0177	END of code.
0178	START of VARIABLE storage.
027A	END of VARIABLE storage.

CREATE - CREATE AN ASCII TEXT FILE
AN MPX-9 UTILITY

This utility allows the user to construct a text file on disk using 'line at a time' entry. Lines are entered and stored directly on disk. Once the user has pressed <carriage return> there is no way to alter a line. Load @ 1000

```
MPX?[N/]CREATE <filespec> [options]
```

This command will fetch the CREATE utility from drive #N, then it will create a file with attributes as described by <filespec>. If extension and/or size is not given, they will default to <.TX> and 8 sectors respectively. If a drive number is not specified as part of <filespec>, the file will be created on drive #1. After the file entry is created in the directory, all sectors of the file will be 'scratched' by filling them with EOT ($04) codes. The user will then be prompted to enter text on a line at a time basis. The current line number will appear on the left. Any editing MUST be done before the user presses the carriage return key. The standard MPX-9 editing commands, ^H and ^X, are the only editing keys supported. When the user is finished, he exits from the editor by entering a line whose first two characters are '/#'. This combination was chosen as it is unlikely to ever be used anywhere else.

OPTIONS:

/X Results in creation of a file with <.EX> extension for use with the -EXEC- utility.

/# Results in the creation of an empty file. There will be no prompts for text entry.

RELATIVE ADDRESS	
0000	START of code.
015E	END of code.
015F	START of variable storage.
02D2	END of variable storage.

DISKEDIT - DISK SECTOR EDITOR
AN MPX-9 UTILITY PROGRAM

This utility allows the use to change the actual contents of a disk sector. The utility is called with:

```
MPX? DISKEDIT(cr)
```

Load at 1000

The screen will erase and you will be prompted for the drive number in which the disk that you would like to edit resides.

You will then be asked which sector you would like to edit. If you enter an invalid number, you will be asked to reenter it. If there is a disk error, you will be given the MPX-9 disk error message, then asked for a new sector number.

After you enter a valid sector number, the program will display the contents of the sector in both HEX and ASCII format on the console device. The contents of the header block are listed on the top line. If the sector is smaller than 256 bytes, only those bytes that actually exist will be displayed. There are several conventions used in displaying the ASCII data. The MSB is masked off, all control characters are represented with a hyphen '-', and positions that are past the byte count for this sector are indicated with '<' symbols.

You will then be asked to enter a command. Any of the following are valid.

F - move forward to the next physical sector.

B - move back to the previous physical sector.

+ - move forward until a forward link of 0 is encountered. This will take the user directly to the end of a file. There is no effect if the current forward link is 0.

- - move back until a reverse link of 0 is encountered. This will take the user directly to the beginning of a file. There is no effect if the current reverse link is 0.

R - move to a sector other than the next/previous physical sector. You will be prompted to enter the sector number that you wish to edit.

M - modify the contents of the current sector.

X - exit from the disk editor program. Exit is via RTS.

N - display the contents of the current sector again. You will be allowed to vector the output to any active I/O device when you use this command.

When you leave a sector by any command except 'N', a check will be made to see if you made any modifications. If you did modify the sector, you will be asked if you wish to save the modified version. Answer the question with a 'Y' or a 'N'. Answering 'Y' will write the modified version to the same sector from which it was read. Answering 'N' will result in keeping the old version.

MODIFICATION: If you enter the 'M' option, you will be asked for the address of the data to be modified. Enter the address of the data byte to be changed by adding the row address (leftmost column) to the column address (top row of single digit numbers). You will be advised of the address and its current contents. You may do several things. Enter a number (hexadecimal, only the last two digits are kept) and press return if you are finished, '^' to go back to the previous byte, and any other character to go to the next byte. If you do not enter a number, the data byte will not be changed. You may enter ASCII characters by typing a single quote instead of a number, then pressing the key that you wish to have inserted.

The header information may be changed. At the address prompt, enter a '>' instead of an address. You may then press F, B, C, or T. They will allow you to change the forward link, reverse link, byte count, or file type respectively. After you press one of these keys, a space will be output and you may enter the new value. YOU MUST ENTER A HEXADECIMAL VALUE HERE. If you simply press a carriage return, a zero will be inserted !

WARNING: Altering the header of a file is a DRASTIC measure, and can have severe implications for the MPX-9 DOS. You should not attempt to do so unless you KNOW what you are doing, and should never do so without having a known good backup diskette.

RELATIVE
ADDRESS 1000

0000	START of code.
05AB	END of code.
05AC	START of variable storage.
06C6	END of variable storage.

EXEC - FILE EXECUTOR
AN MPX-9 UTILITY PROGRAM

This utility allows the user to execute a text file which acts as input to the MPX? prompt without user intervention. It allows for user prompts, and for conditional execution of some commands. See the instructions on the CREATE utility for preparing the command file.

4000

MPX? [N/]EXEC <filename>

The EXEC utility will be fetched from drive #N, and will read the file <filename> line by line, executing each line as it is read. A line may be a comment line, an option line, or a command line. Comment lines begin with '*' symbols. Everything after a '*' will be ignored. Option lines begin with a '/' symbol. The options are discussed below. Command lines consist of MPX-9 commands, given in the same manner that they would be given to an MPX? prompt. Command lines are executed when they are read.

OPTIONS:

// - signifies that the program is finished and control should return to MPX-9 via an RTS.

/B - rings the console bell. May be followed by more commands or options on the same line.

/E - erases the console screen. May also be followed by more commands or options on the same line.

/F - outputs a FORM FEED character. May also be followed by more commands or options on the same line.

/K=@ - where '@' is replaced by any ASCII character. If the input to the last /W option was that character, then any commands or options following on that line will be executed, otherwise the rest of the line will be ignored. MUST BE FOLLOWED BY MORE COMMANDS OR OPTIONS.

/L - will check to see if the filespec specified after the '/L' exists. The result may be obtained by using the /Z option. May be followed by more commands or options on the same line.

/P - will print the rest of the line to the console. A CR/LF will be output if there is nothing else on the line.

/X - chain to another EXEC file. Must be followed by a filespec.

/W - Wait for one of the characters fillowing the '/W' to be pressed. Leading spaces are ignored, internal spaces are considered to be characters. If no characters are specified, pressing any character will allow the user to proceed. The character may be retrieved later with the /K=@ option.

/Z=N - If the last disk operation or command returned a status
/Z>N that satisfies the condition, the rest of the line will be
/Z<N executed, otherwise it will be ignored. MUST BE FOLLOWED BY MORE COMMANDS OR OPTIONS.

A sample program is shown below. The line numbers are not part of the file. They are included for documentation only. Each' must end with a carriage return.

```
 1 * Sample program for EXEC Utility Demonstration.
 2 * Lines 1 and 2 are comment lines for program documentation.
 3 /E /B /P WELCOME TO THE EXEC DOMONSTRATION
 4 * Line 3 will erase the screen, ring the bell, and print a
 5 * welcome message.
 6 /P
 7 /P WOULD YOU LIKE TO LIST THE DIRECTORY OF DRIVE 1 TO THE PRINTER?
 8 /P PRESS 'Y' OR 'N'
 9 /W YN
10 * Lines 6, 7, and 8 will ask the user a question and wait for his
11 * response.  Only keys 'Y' and 'N' will be accepted by line 9.
12 /K=Y LP
13 /K=Y /Z>0 /P ERROR LP NOT LOADED
14 /K=Y /Z>0 //
15 /K=Y PDIR 1 #LP *SAMPLE FROM EXEC PROGRAM
16 * Line 12 will run utility file 'LP.CM' if the key was 'Y'.
17 * Line 13 will advise of error if 'LP.CM' not correctly loaded.
18 * Line 14 will exit the program if 'LP.CM' not correctly loaded.
19 * Line 15 will run PDIR to print the directory.
20 /X EXEC2
21 * nothing after line 20 will ever be reached.
22 * control will transfer to file 'EXEC2.EX'
```

WARNING: 'EXEC' MUST be located in memory so that it doesn't disturb any utilities, drivers, or user programs that it may invoke.

RELATIVE ADDRESS	
0000	START of code.
01EF	END of code.
01F0	START of variable storage.
0386	END of variable storage.

HEXLOAD - LOAD A HEXADECIMAL (S1-S9) FORMATTED FILE
AN MPX-9 UTILITY PROGRAM

This program will allow you to load files produced by assemblers into memory. This particular program will allow you to relocate the code elsewhere in memory. It is called by:

```
MPX? HEXLOAD <filename> [offset](cr)
```

0100

The filename must be specified. The optional offset should be specified in hexadecimal form. This offset will be ADDED to the location at which the file would normally load. Most 6809 programs will be assembled to load at location 0, so the offset will be the actual load address for these programs.

After the hex loader has loaded the file, it will prompt for additional input files. You may continue entering filenames and offsets until you are finished. YOU MUST SPECIFY THE OFFSET FOR EACH FILE, it does not carry over. When you have finished, just enter a return at the prompt, it will exit the hex loader via an RTS.

All error conditions are reported, and then you are prompted for a new file and offset. Be aware that if an error occurs the file has only been partially loaded into memory.

RELATIVE ADDRESS	
0000	START of code.
01E1	END of code.
01E2	START of variable storage.
0326	END of variable storage.

LIST - LIST THE CONTENTS OF A FILE
AN MPX-9 UTILITY

This utility will allow you list the contents of a file to either the console device, or to an alternate device, or to both. The listing may be specified to have decimal line numbers to facilitate editing. The command is called by:

```
MPX? [N/]LIST [#DV] <filename> [options]
```

1000/4000/6000

The [N/] option specifies from which drive to load the LIST utility. The [#DV] option will allow you to vector the output to any device with an active DCB, default output is to the console device. The <filename> should be specified as a standard MPX-9 filename. There are three (3) options available with this command.

/N - will toggle the line number printing function.
normally OFF

/C - will print to the console as well as to the alternate device. normally OFF

/F - will toggle the flag which indicates that the printer accepts harware form feed characters. normally OFF

Each option MUST be preceeded by a "/" mark !

This position independent utility should be located at some high position in memory so that user programs such as BASIC may call it.

NULL COUNT @0132
CURRENT COPY
ON DRIVER disk

RELATIVE ADDRESS	
0000	START of code.
0011	Form Feed Flag
0017	Console Flag
001D	Number Flag The three preceeding flags are normally set to zero which will result in the option defaulting off. Altering the byte to $FF and saving the corrected version will result in that option defaulting ON.
0199	END of code.
019A	START of variable storage.
02C6	END of variable storage.

PATCHES @107 = $2C Channel 12 was $36
@123 = $33 Page Length was $42
@130 - 4F, C6, 20, 8D, 24, 5A, 26, FB
NULLS TO PRINT AFTER LINE FEED (FOR 1200 BAUD)

LPSERIAL - DRIVER FOR SERIAL LINE PRINTER
AN MPX-9 DEVICE DRIVER

This driver will install a DCB named 'LP' for use with MPX-9. It assumes that your printer utilizes an ACIA type interface. The driver is installed by entering the following command:

```
MPX? LPSERIAL(cr)
```

4F80

The line printer driver and DCB are now installed, and will not be removed until your computer is reset or MPX-9 is re-booted from disk.

The driver is written in position independent code, and should be relocated to a section of memory that is set aside for it and will not be overwritten. It occupies $65 bytes of memory, including the DCB.

It is suggested that you rename the device driver -LP.CM- if your system printer is serial. Any program that automatically loads the system printer driver should use the name -LP.CM- so that the end user can select between using a serial or parallel printer driver.

NOTE: This driver MUST be loaded into a location where it will not be disturbed by user programs or utilities.

4F80

RELATIVE ADDRESS	
0000	START of PART 1 code.
0001	Two character device name. ('LP')
0013	16 bit address of ACIA. (Sent as $801C.)
002E	END of PART 1 code.
002F	START of PART 2 code.
0058	END of PART 2 code.
0059	START of DCB.
0062	END of DCB.

4f80
59

4fD9

4F1F

4F80
62

4FE2
30

MEMTEST - A 6809 MEMORY TEST
AN MPX-9 UTILITY PROGRAM

This utility will examine a given range of memory locations and perform a comprehensive check for faulty memory. It is invoke with the following command:

```
MPX? [N/]MEMTST <BEGIN> <END>
>>> <BEGIN> <END>
```

1000

The first line will fetch the utility from drive #N and execute it. All memory locations from BEGIN to END, inclusive, will be tested. A series of '+' characters will be output periodically to assure you that your machine is functioning. When the entire range has been tested, MEMTEST will issue a '>>>' prompt and await further input. You may specify another range of memory locations to test or enter a carriage return to exit back to MPX-9.

When errors are detected, a register dump will occur. The contents of the 'X' register point to the address where the error occured.

CAUTION: Do not test the 4K block in which MPX-9 resides. If you do, the entire system will die.

NOTICE: This memory test is designed to find only those types of errors that cause disk read and seek errors. There are some types of address convergence errors that will not be detected.

RELATIVE ADDRESS	
ØØØØ	START of code.
ØØB6	END of code.
ØØB7	START of variable storage.
ØØCA	END of variable storage.

OUTVECTR - ALTERNATE OUTPUT VECTOR
AN MPX-9 UTILITY PROGRAM

This utility permits the MPX-9 (tm) DOS to send all output that would normally be sent to the console device 'CN' to any other device that has an active DCB. The command is called in two forms:

```
MPX? OUTVECTR #DK:<filename>
MPX? OUTVECTR #LP(cr)
MPX? OUTVECTR(cr)
```

?

The first command will vector all subsequent console output to the disk file named <filename>. This file will remain open until another OUTVECTR command is issued, or until the user writes an End of File character, hex 4, to the disk file. The second command will vector all subsequent console output to the line printer ('LP'). The third form will restore output to the console device ('CN').

CAUTION: When output is rerouted to an alternate device, it is NOT sent to the console, therefore you may not see the MPX-9 advisory messages.

All input will continue to come from the console device unless the INVECTOR command has been used.

RELATIVE ADDRESS	
0000	START of code.
00A7	END of code.
00A8	START of variable storage.
01D5	END of variable storage.

PDIR - PRINT DISK DIRECTORY
AN MPX-9 UTILITY PROGRAM

This utility permits the MPX-9 (tm) disk directory to be printed in a formatted manner to the system console or to any device with an active DCB. The following commands are all allowed:

```
MPX? [N/]PDIR [D] [#DV] [* TITLE FOR LISTING]
MPX? 2/PDIR #LP
MPX? PDIR 2 *LISTING TITLE
```

1000

The first example gives the entire format of the PDIR command line. All items enclosed in [] are optional. The [N/] option will fetch the utility from the disk in drive number N. The [D] option will list the directory from the disk in drive #D, default value of D is 1. The [#DV] option will route the output listing to the device whose DCB identity is given by ´DV´, default device is the current console. The [*TITLE] option may be utilized to print a title over the directory listing for future identification. All characters after the ´*´ are considered to be part of the title. Leading spaces are skipped.

The options may be specified in ANY order, but once the ´*´ is encountered, everything else in the line is considered TITLE.

The utility will list all files, their size, beginning sector umber, end sector number, and their starting address if they have one. Free blocks between files will be shown with their size, beginning sector number, and end sector number. At the end of the listing, PDIR will indicate the next free sector on the disk, and the total number of USED sectors.

RELATIVE ADDRESS	
0000	START of code.
031E	END of code.
031F	START of variable storage.
032E	END of variable storage.

VERIFY - VERIFY THE READABILITY OF A DISKETTE
AN MPX-9 UTILITY

This utility will examine a given range of sectors on the diskette and determine their status. Non-existant sectors, ending sectors, and all sectors shorter than 256 bytes will be FLAGGED. This utility is invoke with the following command:

```
MPX? [N/]VERIFY <D>
MPX? [N/]VERIFY <D> [BEG]
MPX? [N/]VERIFY <D> <BEG> [END]
```

These commands will fetch the VERIFY utility from drive # N, and proceed to read sectors. The first command will automatically set the first sector to 0 and the ending sector to 399. The second command will allow you to specify the first sector, and automatically set the last sector to 399. The third command will allow you to specify both first and last sectors. The default value of 399 as the last sector is found at the beginning location of the utility + 7. It is a 16 bit hexadecimal number. You may alter this value and resave the utility if desired.

RELATIVE ADDRESS	
ØØØØ	START of code.
ØØØ7	Default ending sector number, decimal 399 (hex $18F).
Ø1Ø2	END of code.
Ø1Ø3	START of variable storage.
Ø2Ø7	END of variable storage.