REX

PT68K4/PT68K5

User's Manual

PALM BEACH SOFTWARE
12364 CR 223
OXFORD, FL 34484

INTRODUCTION

REX is a single user disk operating system for the 680XX microcomputer. The syntax and overall look and feel is based on a disk operating system from Digital Equipment's PDP-11 series. Any resemblance to other operating systems is purely coincidental.

There are 3 divisions to REX: the Disk Operating System, the File Management System and the Utility Command Group. Each of these will be explained in its own section.

REX version 4 is for the PT-68-K2/K4 and version 5 is for the CD-020. The only difference between the two is the addresses of the printer drivers and the floppy disk drivers. The convention of REXK? will be used throughout this manual and you should use the version supplied on the system disk. Instructions for REXK4 which are different than for REXK5 will be in parentheses ( ).

Getting STARTED

The System disk which is supplied for the CD-020 is a HD disk prepared for either a 3.5" 1.4 meg drive or a 5.25" 1.2 meg drive. If you don't have that type of floppy drive you must make a special request for the proper disk. The drive must be addressed as Drive #0 and connected to the 37C65 disk controller. The System disk which is supplied with the PT-68-K2/K4 is a 720K disk in either 3.5" or 5.25" size. The drive must be addressed as Drive #0 and connected to the 1772 disc controller. Drive #0 is the end connection on the IBM type 34 pin cable, after the twist. The jumper on the drive must be in the second drive position.

When you turn the computer on you should have "MONK:" displayed on the screen, if you don't check the Monk manual. Insert the system disk in drive #0, be sure the write protect tab is in place, and press the "V" key to load REX. Unless your clock has been set you will be asked to enter today's date as Month, Day, Year; do so and you should have "REX:" displayed on the screen.

At this point a number of the Utility routines will work. You should try CAT, this will list all the programs on the system disk. Other routines that you might like to try are SETIME, DATE, YEAR, and ASN. The next task is to make a backup copy of your system disk; however, we must have two drives in order to make the COPY utility work. The easiest way to do this is to create a RAM disk.

Check to see if the memory end pointer is set to $6FFFF($FFFF), type MEMEND and the current default value will be displayed. Type MEMEND,6FFFF(FFFF) to set it, then type MEMEND again to see if it changed. Next CAT,7 to see if the RAM disk has been installed, if not then type TURBO,7 to install it. Copy the System disk to the RAM disk with COPY 0,7. When this is finished you must now create a formatted disk. Put a brand new disk in drive #0 and type 7.FORMAT,0. Give the disk a name of SYSTEM and and number of 11, answer the question with a "Y". After the disk is formatted type 7.COPY 7,0. When the copy is complete type LINK 0.REXK?F3.COM for a 3.5" diskette or LINK 0.REXK?F5.COM for a 5.25" diskette. You should now have a working System Disk. It

would be a good idea to make several more copies. The Utility routines which you have just used are explained in the Utility Command Section.

The Syntax of a REX command is:
[Drive #<.>]<Filename>[<.>Extension][<,>Parameters]

For example; 0.RENAME.COM,1.FILE1.TXT,1.FILE2.TXT
This is the equivalent; RENAME,FILE1,FILE2
The items enclosed in [ ] are optional while items enclosed in < > are essential. In the above example, if the Drive # is used the <.> must follow the number. If the drive # is not included then REX will use the System Drive # (usually drive #0). The Filename is necessary but the Extension is not; if an Extension is used, it must be preceded by the period. If an extension is not used then an extension of .COM is assumed for the first filename in a command. The first parameter filename of 1.FILE1.TXT can be shortened to FILE1. REX will use the working drive # if a drive # is not specified and the RENAME utility will default to a .TXT extension if it is missing. The second parameter works about the same. The Space is the same as the comma (,) in the file specification so you can use whichever is more convenient. e.g RENAME FILE1 FILE2.

The [Drive #<.>]<Filename>[<.>Extension] is called a File specification or filespec for short. Several filespecs will make a Command. Normally there is one Command on a Command Line, but 2 or more could be there if they are separated by the colon (:). See the EXEC utility for an example.

There are two memory resident commands in REX. The first is MON which will return control to MONK, the ROM resident monitor. The second is GET which will load a binary type file but will not run it. The Syntax for GET is: GET,<filespec>.

At this time you need to tell REX what types of Floppy or Hard drives you have connected to the Computer. This is done thru the Utility DRIVESET (please go read the instructions for DRIVESET 3 times.) REX can access up to 10 (0 to 9) Logical Drives. The information that describes these drives is kept in the Drive Info Table (DIT) at $3B00. The amount of memory that you have in your system is a factor in setting the RAM disks as well as the Track buffers. I suggest that you read through the rest of this manual before you make your decisions.

After REX has booted up and displays or get the date it looks for a file called UPSTART.TXT. If it finds this file it will then execute the contents as though the input was from a keyboard. Assuming that you have got a DRIVE(X), a PRINT(XX) and a TURBO,X that you need to install each time you boot REX you need to put these in the UPSTART.TXT file. Now every time you boot REX these programs will be run and REX will be ready to go. A sample UPSTART.TXT would look like this:

```
PRINTAB:DRIVEA:TURBO,7:WORK,1
```

The UPSTART provided on your disk is:

LOGO:SSBANER for REXK5
MEMEND,FFFF:MEMEND:TURBO,7 for REXK4

Since REX comes up with system and work drives set to drive #0, the Printer driver would load from drive #0. DRIVEA would also load from that drive. TURBO,7 and WORK,1 would load from what ever Drive #0 that you specified in DRIVEA. If you have a hard disk you might want to specify HIDE(HARD) as the first entry in UPSTART and then DRIVEH. If H0 is now L0, the rest of the entries would load from the Hard drive a much faster process.

If you want to boot from the Hard drive you must prepare a version of REXDOS that loads the HIDE(HARD) along with REX. e.g APPEND,REXK?I.COM,HIDE.COM,REXDOS.SYS. Copy REXDOS to the Hard disk and link it. Make sure that the DRIVEH is on the hard disk and any other routines that you will put in UPSTART. Create an UPSTART.TXT with EDD, put it on the Hard disk, go to MONK (MON) and enter "W". Now that's a fast boot.

REX supports two printers #1 and #2. These can be either parallel at PORTA, PORTB, or Serial at COM1, COM2, COM3, COM4. Once the printer drivers are set to your hardware selection and the printer connected, you direct output to the printer by putting a "P" before the command. e.g. P,CAT would output the catalog of the working drive to printer #1. There are a number of "P" commands PS will print small (16 cpi) on #1 while PS2 will print small on #2.

THE DISK OPERATING SYSTEM

The DOS has a number of user accessible variables and subroutines. The addresses of these routines are in a file called REXEQU.TXT, which is supplied.

GLOBAL VARIABLES

$2300	BSPCHR	Backspace character
$2301	DELCHR	Delete character
$2302	EOLCHR	End of line character
$2303	DEPTH	Screen display line count
$2304	WIDTH	Screen display max column count
$2305	NULLS	Pad count for teletype terminal
$2306	TABCHR	Tab character
$2307	BSECHR	Backspace echo character
$2308	EJECT	Page eject count
$2309	PAUSE	Pause control count
$230A	ESCCHR	Escape character
$230B	SDRN	System drive number
$230C	WDRN	Working drive number
$230D	SYSFLG	Use system drive flag
$230E	SYSMON	System month
$230F	SYSDAY	System day
$2310	SYSYER	System year
$2311	LSTTRM	Last line terminator character
$2312	USRCMD	User command table pointer
$2316	LBPTR	Line buffer pointer
$231A	ESCRET	Escape return pointer
$231E	CURCHR	Current character
$231F	PREVCH	Previous character
$2320	CURLCT	Current line number
$2321	XFRFLG	Transfer flag
$2322	TFRADR	Transfer address
$2326	OFFLAD	Loader offset address
$232A	ERRTYP	Error type
$232B	IOFLAG	Special I/O flag
$232C	OUTSWT	Output switch
$232D	INSWT	Input switch
$2336	DOCMDF	Command flag
$2337	CURCOL	Current output column
$2338	MEMEND	Memory end pointer
$233C	LODADR	Binary load address

USER CALLABLE DISK OPERATING SYSTEM ROUTINES

$2400	COLDS	Cold start address.
$2404	ARMS	Warm start address.
$2408	RENTER	Re-enter main loop.
$240C	INCH	Input character. (user changeable)
$2410	INCH2	Input character.
$2414	OUTCH	Output character. (user changeable)
$2418	OUTCH2	Output character.
$241C	GETCHR	Get character.
$2420	PUTCHR	Put character.
$2424	INBUFF	Input into buffer. Input from the keyboard to the LINBUF. The delete character, backspace character, CR and LF are recognized, all other control codes are ignored. The cancel (CTL X) will prompt (???) and reenter INBUFF. The backspace will go back to the beginning of the buffer but not before it. The LF will echo the CR/LF to the terminal but only place a space in the buffer. The CR will terminate input and will be placed in the buffer. A maximum of 128 entries are allowed. On exit the LBPTR is pointing to the first character in the Line buffer.
$2428	PSTRNG	Print string. A1 points to the first character of string. A CR/LF is output and then the string is output until a $04 is encountered. On exit the A1 register is pointing to the next character after the $04, all other registers except D0 are preserved.
$242C	CLASS	Classify character. The character in D0 is tested and if it is an alpha or numeric character the carry is cleared on return. Any other character and the carry is set. All registers are preserved.
$2430	PCRLF	Output CR/LF.
$2434	NXTCH	Get next character. The character in CURCHR is moved to PREVCH and the character pointed by LBPTR is put in CURCHR and D0. More than one space is ignored and the LBPTR is advanced to point

to the next character unless the last character was a CR or EOL. Exit is through CLASS and carry will be clear if character is alpha-numeric. All registers are preserved except D0.

$2438 RSTRIO Restore I/O vectors.
This routine will set a number of I/O vectors and pointers to their original values. OUTCH is reset to OUTCH2, INCH is reset to INCH2, PAUSE is set on, INSWT, OUTSWT, and IOFLAG are all set to zero.

$243C GETFIL Get file specification.
A0 must point to the FCB and LBPTR must point to a file specification. The file spec is parsed and the drive #, file name and extension, if any, are put in their proper places in the FCB. On exit the carry will be clear if no errors were discovered and LBPTR will be pointing to the first character beyond the separator unless the separator was a CR or EOL. All registers except D0 & D7 are preserved.

$2440 LOAD Load file.

$2444 SETEXT Set default extension.
A0 should point to the FCB where you want to install the default extension. D0 to contain the code character of the extension. If the file specification does not already have an extension, the extension code in D0 will be installed. All registers are preserved. The following are legal extension codes, all others are ignored.

0 - BNY	6 - SRC	12 - RCM
1 - TXT	7 - DAT	13 - REL
2 - COM	8 - BAC	14 - MUS
3 - BAS	9 - DIR	15 - PIC
4 - SYS	10 - PRT	16 - GIF
5 - BAK	11 - OUT	

$2448 OUTDEC Output decimal number.
Will output as a decimal number the binary number contained in D0. If D1 is = $FF then the decimal number will be output without any leading zeros or spaces. If D1 = 0 then 9 digits will be output with leading zeros as spaces. If D1 = 1 thru 9 then 9-n digits will be output with leading zeros as spaces. The least significant digit will always be output. All registers except D0 are preserved.

$244C OUTHEX Output hexadecimal number.
Will output 2 hex digits from a single byte pointed to by the A1 register. On exit A1 will be pointing to the next byte, D0 will contain the last character printed, and all other registers are preserved.

$2450	RPTERR	Report error. See the Error Number List.
$2454	GETHEX	Get hexadecimal number. Will read a hexadecimal number from the command line pointed to by LBPTR into D0. Reading will stop on the first non-hex character encountered. Return with NEG set indicates that no entry was made, carry will be set if any character, before the separator, was not a valid hex character. Carry will be clear if a valid hex number was found. All registers except D0 & D7 are preserved.
$2458	OUTADR	Output address. A1 points to a 4 byte address which will be output as 8 hex digits. On exit A1 points to the last byte plus one. All registers preserved except A1 & D0.
$245C	INDEC	Input decimal number. Will read a decimal number from the command line, pointed to by LBPTR, into D0 as a binary number. Reading will stop on the first non-decimal digit encountered. Return with NEG set if no entry, carry will be set if any character, before the separator, was not a valid numeric character and LBPTR will be advanced to the next separator character, which will be in D0. Carry will be clear if a valid decimal number was found. All registers except D0 & D7 are preserved.
$2460	DOCMND	Call REX as subroutine.
$2464	STAT	Keyboard check. The status of the keyboard is checked and if busy, not equal zero is returned. All registers except D7 are preserved.
$2468	INCHNE	Input character w/o echo.
$246C	PDATA	Print string w/o CR/LF.

PRINTER ROUTINES

$2F00	PINIT	Initialize printer #1.
$2F04	PCHEK	Check if printer #1 busy.
$2F08	POUT	Output to printer #1.
$2F0C	PINIT2	Initialize printer #2.
$2F10	PCHEK2	Check if printer #2 busy.
$2F14	POUT2	Output to printer #2.

FILE MANAGEMENT SYSTEM ROUTINES

The File Management System is the link between the DOS and the Disk. It takes care of all the housekeeping without any concern of the user. Access to the FMS is through the File Control Block (FCB). A FCB is a block of 320 bytes of memory and is addressed as follows:

Byte	Description
0	Function Code
1	Error code
2	Status
3	Drive number
4-11	File name
12-14	Extension
15-16	MSW of link pointer
17-16	Starting File address
19-20	Ending File address
21-22	File size
23F	File Sector Map
24	Assigned drive #
25-27	File Creation Date
28-29	FCB link pointer
30-31	Current T&S
32-33	Current Record #
34	Data Index
35	Random Index
36-46	Name Buffer
47-49	Current Directory Address
50-52	First Deleted Directory Pointer
53-63	Scratch
59	Space Compression Flag
64-319	Sector Buffer

ENTRY POINTS TO FMS

$3004	FMSCLS	Close all files. Restore the I/O vectors and clear the SIR TABLE
$3008	FMS	Call FMS. A valid function code must be in Byte 0 of the FCB and A0 must be pointing to the FCB. The following Function Codes are valid.

FC	DESCRIPTION
0	Read or Write next byte
1	Open for Read
2	Open for Write
3	Open for Update
4	Close file
5	Rewind file
6	Open Directory
7	Get Information Record
8	Put Information Record
9	Read Single Sector
10	Write Single Sector
11	Extend Directory
12	Delete File
13	Rename File
15	Next sequential sector
16	Open System Information Record
17	Get Random Byte from Sector
18	Put Random Byte in Sector
19	Open & Extend file
20	Find next drive
21	Position to Record N
22	Backup one Record

REGISTER USAGE

In any calls to FMS, A0 must be pointing to the FCB. A0 is preserved and if you don't need it for something else, you can set it once and forget it.

A1 is not preserved. It is the source register, it must be pointing to the data to be output etc. It will also be used to output error messages.

A2 thru A6 are always preserved in any calls to REXDOS.

D0 is used to pass data to and from REXDOS. It is not preserved.

D1 is used as an error flag and modifier; it is not preserved

D7 is scratch and not preserved. D2 thru D6 are preserved.

USER COMMAND TABLE

Memory-resident routines can be accessed thru the User Command Table. The command table address is put at USRCMD ($2312).

The file spec in the user command table must be 8 bytes long, the 8th byte must be $0. For example;

```
DC.B    'REW',0,0,0,0,0
DC.L    <VECTOR TO REW>
DC.B    'RENAME',0,0
DC.L    RENAME
DC.B 0  Terminate table with $0
```

UTILITY COMMAND GROUP

The following conventions will be used in describing the Utilities and the Syntax to use them.

< >	Mandatory entry
[ ]	Optional entry
( )	Do not enter, inserted only for clarity
,	necessary but can be replaced by a space
:	necessary to separate 2 or more commands on a command line.
X	Any alpha or numeric character
filename	The name of a file or utility
filespec	The drive #, filename, and extension of a file or program
drive	The drive # to be used
drive list	A list of one or more drive # separated by comas
match list	A set of names or extensions which will limit the execution of the program to only those files which match the names in the match list
command	A collection of filespecs and other data to instruct REX to do something
command line	A list of one or more commands

Most Utilities will display a help message if the syntax is wrong of if a (?) is substituted for the filespec or drive #.

ACAT

Alphabetized LIST OF DIRECTORY 4 COLUMNS WIDE

Syntax:ACAT<,drive>

APPEND

CONCATENATE TWO OR MORE FILE TO MAKE ANOTHER FILE

Syntax:APPEND,<filespec>[,filespec],<filespec>

The last filespec is the output file and will contain all of the preceding files.

ASN

REPORT OR ASSIGN THE SYSTEM AND WORK DRIVES

Syntax:ASN[,W=<drive>][,S=<drive>]

Without any arguments ASN will report the system and work drive numbers. With either or both arguments it will assign new drive number to the work or system drive numbers. See SYSTEM & WORK.

BNY2S123

MAKE A MOTOROLA FORMAT FILE FROM A REX BINARY FILE

Syntax:BNY2S123,<filespec>

Depending upon the load address of the file. the S1, S2 or S3 format will be used. If a Transfer address is in the file then an S7, S8 or S9 record will terminate the file.

BORDERC

SET THE BORDER COLOR

Syntax:BORDERC,[color #]

The default color # is 0(no color). Other colors are 1=blue, 2=green, 3=light blue, 4=red, 5=magenta, 6=yellow and 7=white.

CASHON (REXK5 only)

TURN CACHE ON

Syntax:CASHON The Cache will be turned on if available.

CASHOFF (REXK5 only)

TURN CACHE OFF

Syntax:CASHOFF The Cache will be turned off.

CAT

LIST THE DIRECTORY OF SELECTED DRIVE

Syntax:CAT[,<drive list>][,<match list>]

The default drive is Work drive and if no match list is entered then all files on the drive will be listed. The match list will limit the listing to only the files which are matched. A number of options are available; if you enter CAT + a help message will be displayed and the options listed with the current settings. The options are:

A Alphabetize the listing.
D Display the creation date of file
F Display the File number.
M Display with F,S,D options.
N Include non-existant file in listing
P Paging
R Repeat
S Display file size.

For example: CAT+ADF,1,.COM would display all the .COM files on drive #1 in alphabetical order along with the file number and the creation date.

CD

WILL INSTALL THE SELECTED RESOURCE LIST DRIVE OR PARTITION ON THE LOGICAL DRIVE NUMBER SELECTED

Syntax:CD,<Resource #>,<Logical Drive #>

The Resource # can be hexadecimal if preceded by the dollar ($) sign, else enter the decimal number. The Logical Drive # should be a decimal (0-9).

CGS

CLEAR THE GRAPHICS SCREEN

Syntax:CGS,[color #]

The Graphics Screen will be cleared
The default color # is 1=blue. Other colors are 2=green, 3=light blue, 4=red, 5=magenta, 6=yellow and 7=white.

CHECK

CHECK TO SEE IF TWO FILES ARE IDENTICAL

Syntax:CHECK,<filespec>,<filespec>

The program will report that either the two files are identical or that they do not match.

CMPBNY

COMPARE TWO BINARY TYPE FILES

Syntax:CMPBNY,<filespec>,<filespec>

The two files will be compared and any differences will be reported on the screen.

CMPMEM

COMPARE A FILE AGAINST MEMORY

Syntax:CMPMEM,<filespec>

The file is checked against memory and any differences are reported on the screen.

COPY

COPY FILES FROM ONE DRIVE TO ANOTHER DRIVE

Syntax:COPY[,X],<filespec>,<filespec>
:COPY[,X],<filespec>,<drive>
:COPY[,X],<drive>,<drive>[,matchlist]

There are three different forms of the Copy utility, and in addition there are a number of options which can be set. The options are:

A Copy in alphabetical order
C Allow corrupt files to be copied
D Copy files with a newer date
E Delete the existing destination file
F Copy files by number
K Kill duplicate file on source
L List files without copying
N Copy files not on the destination drive
O Turn off the default options
P Prompt before copying
R Recover file from Track-Sector
S Make a second copy of file
U Use the current DOS date
W Wait for disk change
Z Zap source file after copying

For example: COPY,DL,0,1,.SRC would list the files with a later date and a .SRC extension on drive #0, that are on drive #1.

CS

CLEAR THE SCREEN

Syntax:CS

CURSORC

CHANGE THE CURSOR COLOR

Syntax:CURSORC,[color #]

The default is 77=white. Other colors are 11=blue, 22=green, 33=light blue, 44=red, 55=magenta and 66=brown

DATE

DISPLAY OR CHANGE THE SYSTEM DATE

Syntax:DATE[,MM,DD,YY]

For example: DATE will display the current system date. DATE,01,01,90 Will set the system date to Jan 1, 1990.

DELETE

DELETE FILE FROM DISK

Syntax:DELETE,<filespec>

There is no default extension; you must include an extension in the filespec. The filespec will be displayed on the screen and you will be given the opportunity to abort.

DISKNAME

SERVICE THE DISK NAME, EXTENSION AND CREATION DATE

Syntax:DISKNAME,<drive>

The current name, extension and creation date of the disk in the selected drive will be displayed. You have the option to change it. If you answer "Y" you are asked for the new name, extension, and date. No entry (C/R) will keep the current entry.

DM

DRIVE MANAGER

Syntax:DM

The Data file DM.DAT on the System Drive will be loaded. The contents of the DIT will be displayed with the comments from the Data file. Next the first 9 lines of the Data file will be displayed and you may move the long cursor (BAR) with the up & down arrow keys. If you move to the bottom of the screen the display will scroll up till you reach the end of the Data file. As you move the BAR up you will scroll backwards to the beginning.

In the middle of the screen is a command line with the available commands displayed. The Capital letter in parenthesis is entered to enter the command functions which are described as follows:

Delete The resource entry under the BAR will be deleted.

Edit — The 8 byte DAT will be displayed one at a time in hexadecimal notation. A C/R will leave the data unchanged and advance to the next byte. Entry should be in hexadecimal; if you make a mistake, continue and type in the correct data. Only the last two entries will be accepted. The resource # can't be changed with Edit.

Install — The resource under the BAR will be moved to the Logical drive # selected. The DIT displayed at the top will be updated to reflect this change.

Message — You must enter a 40 byte Message which will be inserted to the resource under the BAR. Fill the rest of the line with spaces to complete the entry.

New — Create a new resource entry. You must enter a hexadecimal #. A new resource line will be inserted in numerical order and the Data file will be redisplayed starting at the beginning.

Print — A copy of the Data file will be sent to the printer.

Quit — Return to the DOS with the changes to the DIT only.

Save — Any changes to the Data file will be saved. You will have the opportunity to abort by not deleting the current Data file.

Using Drive Manager

Enclosed is a copy of my Data file. It includes 59 entries, 6 floppy drives, 3 RAM disks, 9 partitions on an MFM 40 megabyte Seagate ST251 hard drive, and 38 partitions on an IDE 130 megabyte Seagate ST3145A. Note that there are 2 entries to tell me the last cylinder on each hard drive. These entries are not to be formatted.

The 8 bytes in hexadecimal on the left is the data that goes into the Drive Information Table (DIT). This tells REX how to address the drive or partition. See DRIVESET utility for a discussion of these entries. The rest of each line is whatever you want. I make the first entry the decimal number of the resource number. It is easier for me to keep track of decimal numbers. The next entry reminds me of the type of hard drive and the first and last cylinders of the partition. If you don't put this information here you had better put it some place because you will certainly need it again. Next I enter a description of the contents of the partition. As the partitions are filled with files, I use DISKNAME to change the name of the partition to match the contents and the volume # to match the Resource #. Now when I do a CAT,5 the heading display will tell me the title and resource # of drive 5. I could also enter DM and see all the drives displayed on the screen.

The Resource list should be prepared before you format your hard drive. Keep in mind that you can easily make a large partition out of 2 or 3 small ones, so when in doubt make more partitions. To format a hard drive I keep the system drive on zero and fill logical drives 1 thru 9 with sequential resources (24-32). Now format 1 thru 8; 9 will supply the stop cylinder for #8. Next move 32 thru 40 to 1 to 9 and format 1 thru 8. Notice that I

have 2 resources numbers 19 and 63 which are used to set the ending cylinders for the previous entries.

I have both MFM and IDE hard drives on my system. There is a combined driver HRIDE.COM that will handle both drives. This should be appended to REXK?.COM and a DIT to produce the REXDOS.SYS.

An empty DM.DAT file is included on your disk. If you lose it or need to create a new Data file, enter DM and add a message to the first line of zeros; now save it and you will have a DM.DAT on the system disk. Enter DM again and then (N)ew number 32, then New 30 and then New 1,2,3 etc.

Available Drive Resource List

```
00 00 00 03 01 14 12 1C       5.25" HD 1.2Meg Phy #0
01 00 01 03 81 10 12 22       3.5" HD 1.4Meg Phy #1
02 01 07 20 20 FF 0A 12       5.25" DS 720k Phy #0 80 Tks
03 01 07 21 21 FF 0A 12       5.25" DS 720k Phy #1 80 Tks
04 01 07 22 22 FF 0A 12       5.25" DS 360k Phy #2 40 Tks
05 01 07 23 23 FF 0A 12       3.5" DS 720k Phy #3 80 Tks
07 02 00 10 00 00 00 20       2 Meg RAM DISK @$100000
08 02 00 10 00 00 00 30       3 Meg RAM DISK @$100000
09 02 00 30 00 00 00 10       1 Meg RAM DISK @$300000
0A 80 22 CC 00 CC 00 00       10 MFM 1-100 SYSTEM DISK
0B 80 22 CC 00 CC 00 64       11 MFM 100-199 PARASOL, NAMES, ETC
0C 80 22 CC 00 CC 00 C8       12 MFM 200-299 K5 SOURCE CODE
OD 80 22 CC 00 CC 01 2C       13 MFM 300-399 K4 , REX SOURCE CODE
0E 80 22 CC 00 CC 01 90       14 MFM 400-499 EMPTY
0F 80 22 CC 00 CC 01 F4       15 MFM 500-599
10 80 22 CC 00 CC 02 58       16 MFM 600-699 EMPTY
11 80 22 CC 00 CC 02 BC       17 MFM 700-799 APN,HI_TECH 1990 DATA
12 80 22 CC 00 CC 03 20       18 MFM 800-820 TEST PURPOSES
13 80 22 CC 00 CC 03 34       19 MFM 820-820 LAST CYLINDER
14 A0 22 FE 01 FE 00 00       20 IDE 0-49 SYSTEM
15 A0 22 FE 01 FE 00 32       21 IDE 50-141
16 A0 22 FE 01 FE 00 8E       22 IDE 142-234
17 A0 22 FE 01 FE 00 EB       23 IDE 235-284 REX SOURCE CODE
18 A0 22 FE 01 FE 01 1D       24 IDE 285-334 K4 SOURCE CODE
19 A0 22 FE 01 FE 01 4F       25 IDE 335-384 K% SOURCE
1A A0 22 FE 01 FE 01 81       26 IDE 385-434 PARASOL, NAMES, ETC
1B A0 22 FE 01 FE 01 B3       27 IDE 435-484
1C A0 22 FE 01 FE 01 E5       28 IDE 485-534
1D A0 22 FE 01 FE 02 17       29 IDE 535-584
1E A0 22 FE 01 FE 02 49       30 IDE 585-634
1F A0 22 FE 01 FE 02 7B       31 IDE 635-684
20 A0 22 FE 01 FE 02 AD       32 IDE 685-705
21 A0 22 FE 01 FE 02 C2       33 IDE 706-726
22 A0 22 FE 01 FE 02 D7       34 IDE 727-747
23 A0 22 FE 01 FE 02 EC       35 IDE 748-768
24 A0 22 FE 01 FE 03 01       36 IDE 769-790
25 A0 22 FE 01 FE 03 17       37 IDE 791-801
26 A0 22 FE 01 FE 03 22       38 IDE 802-811
27 A0 22 FE 01 FE 03 2C       39 IDE 812-821
28 A0 22 FE 01 FE 03 36       40 IDE 822-831
29 A0 22 FE 01 FE 03 40       41 IDE 832-841
2A A0 22 FE 01 FE 03 4A       42 IDE 842-851
2B A0 22 FE 01 FE 03 54       43 IDE 852-861
2C A0 22 FE 01 FE 03 5E       44 IDE 862-871
2D A0 22 FE 01 FE 03 68       45 IDE 872-881
2E A0 22 FE 01 FE 03 72       46 IDE 882-891
2F A0 22 FE 01 FE 03 7C       47 IDE 892-901
30 A0 22 FE 01 FE 03 86       48 IDE 902-911
31 A0 22 FE 01 FE 03 90       49 IDE 912-921
32 A0 22 FE 01 FE 03 9A       50 IDE 922-931
33 A0 22 FE 01 FE 03 A4       51 IDE 932-941
34 A0 22 FE 01 FE 03 AE       52 IDE 942-951
35 A0 22 FE 01 FE 03 B8       53 IDE 952-961
36 A0 22 FE 01 FE 03 C2       54 IDE 962-971
37 A0 22 FE 01 FE 03 CC       55 IDE 972-981
38 A0 22 FE 01 FE 03 D6       56 IDE 982-991
39 A0 22 FE 01 FE 03 E0       57 IDE 992-1001
3F A0 22 FE 01 FE 03 E9       IDE 1001-1001 LAST CYLINDER
```

IDE Explanation

The IDE drives are getting quite large and present some unique problems for REX. The maximum partition is about 15 megabytes and in a 560 meg drive this is at least 38 partitions. REX can only address 10 drives at a time so a Drive Manager (DM) has been written to allow you to manage a large number of disk resources. See separate instructions for CD and DM; also refer to HIDE and IFORMAT.

The drive information table (DIT) is the first item to figure out. REX will put the IDE drive into translation mode which on older drives was 17 sectors per track and the number of heads specified in the data sheet. The number of heads times the sectors per track is the sectors per cylinder. Keep in mind that one of their sectors is 512 bytes which is two of REX's sectors of 256 bytes each. REX sectors 1 & 2 map to sector #1 etc. If you multiply the sectors per cylinder by the number of cylinders times 512 you should get the formatted capacity of the IDE drive in bytes.

The next item to set is the logical sectors per track. This refers to REX, and has nothing to do with IDE parameters. The maximum value is 254 ($FE) and it must be an even number. I have always tried to make it a multiple of the sectors per track but that is not necessary. REX uses track 0, sector 0 in the linkage bytes to indicate end of file ($0000). Sector 0 is not allowed and that is the reason we can't have 256 sectors per track.

The partition offset is the starting cylinder number of the partition. This is usually ($0000) for the first partition and the next partition offset determines the size of the first partition. IFORMAT will scan the DIT to find the smallest offset larger than the starting offset and use that cylinder number to limit the size to format. You can over-ride this by entering the ending partition number when requested. If you are formatting a large partition close to 65,000 and IFORMAT reports that the total sectors are less than you expected you have exceeded the limits and need to reformat with less cylinders.

The Type Flag should be A0 for a master drive and B0 for a slave drive. The drive number ($FF if not installed) is not used by the IDE driver; however it is used by the Drive Manager (DM) and the Change Drive (CD) utilities.

A sample DIT entry would look like this:

```
$14 $A0 $72 $FE $06AE $0000
```

The $14 is drive #20
The $A0 is Master IDE drive
The $72 is 114 sectors per track
The $FE is 254 logical sectors per REX track
The $06AE is 1710 sectors per cylinder
The $0000 is a starting cylinder of 0

In the above example the maximum number of cylinders would be 37 so the ending cylinder would be 37 (0-36). You could try 38 and see what would happen.

The HIDE driver works like this. When it's loaded it reports that it is installed and jumps to the cold start of REX. The first time an IDE drive is selected the initialization is called. The sectors per cylinder are divided by the sectors per track to get the number of heads. If this

is not even a drive set error will be reported. The Set Parameters command is issued with the number of heads and the sectors per track. This sets the geometry of the drive and can't be changed. If you have a slave drive it must also use the same parameters.

Now that the drive is initialized, the Track & Sector address is calculated by a seek routine. The Track is multiplied by the logical REX sectors per track, and the sector is added to get the number of sectors from the beginning to the addressed sector. This number is divided by the sectors per cylinder to get the cylinder number, then the offset is added to get the cylinder high byte and cylinder low byte. The remainder from this division is divided by the sectors per track to find the head # and this remainder is divided by 2 (their sectors are 512 bytes long) to get the actual sector to address.

REX has to read or write 512 byte sectors to the disk so it puts the data in a buffer and keeps track of what is in the buffer so that only one physical read or write is necessary for 2 REX sectors. The IDE is very fast compared to the older MFM drives; however it is not as fast as the drivers that load the data directly into the destination address. REX puts a sector into a File Control Block (FCB) and then issues 1 byte at a time to the destination address. A Quick Read driver (QLOAD) is available that will more than double the transfer rate by ignoring the REX error checking and placing the data directly into memory. This works very well for graphics but is not recommended for programs or data.

A combined disk driver for MFM and IDE (HRIDE) is available to use both drives on one system. If you have a lot of data on a MFM drive and want to transfer it to the IDE it might be a lot easier to move from HARD to IDE than to copy to floppies.

If you want to boot from the IDE drive you must create a system disk in a partition, have this partition in drive 0, and link Rex. From MONK select the IDE boot command (I for K4, W for K5). Monk will now execute the Identify Drive command to get the number of heads. It will set the Parameters to the number of heads and 17 sectors per track. Next sector 1 on cylinder 0 will be read into the boot buffer ($FF800 on K4, $1000 on K5). The IBM partition table #4 will be examined for an offset, if not zero then the DIT from the Partition Table #4 (Buffer+500) will be used to get Sector 1, Track 0 of the offset cylinder number into the buffer. The first 2 bytes in the buffer are the link (Tk & Sec) of REX, the next 8 bytes are a DIT which will be used to load REX. If your IDE has a different number of sectors per track than 17 (34 REX sectors) I would suggest that you format 1 cylinder using 17 sectors per track, the number of heads times 34 as the sectors per cylinder and a factor of that number as the logical sectors per track. You need a minimum of 2 tracks, 1 track for the directory and at least 1 track for data. Put REXK?Q on this partition. Link REX and see if it will boot. If it does then the version of REXK?I needs to have a DIT appended to it that has the System partition as drive 0 and HIDE appended at the end. Put this version on the 1 cylinder partition and link it.

Keep in mind that if you have 57 sectors per track that 40 sectors will be lost on each track. I do not have one of the larger IDE drives to test so all this is theoretical. The first time you use the IDE drive after REX is booted the Set Parameters will be using the data in the DIT so data in the DIT must match the correct drive parameters. The 1 cylinder partition should start a cylinder 0 and end at cylinder 1, the System partition should start at cylinder 1 and end at 10 or so. Monk will probably have to be changed to read physical sectors per track from the identify drive command to make this easier.

DRIDE (REXK4 only)

Syntax:DRIDE

Combined Disk driver for both MFM and IDE drives.

DRIVESET

DRIVESET Will service the Drive Information Table (DIT).

Syntax:DRIVESET[,(X)]

If DRIVESET (no (X)) is entered, the DIT currently in REX will be loaded. If a character (0-9,A-Z) is entered after a comma (,) then DRIVESET will load a file DRIVE(X) from the Work Drive. If a (?) is entered then a Help message will be displayed.

DRIVESET will ask for a Logical Drive #. Enter the desired number and press the Return key. A mask will be displayed on the screen, and depending upon the type of drive, will display the current drive info as normal bright characters. The low intensity characters will describe the data for the selected drive. A 5.25" high density drive will appear as follows:

00:LOGICAL DRIVE #. ($FF = DESELECTED)
00:TYPE FLAG. (A0=HARD, 0=QUAD, 1=FAST, 2=RAM
00:DRIVE # (0 - 1)
03:SRT STEP RATE (1 ms INCREMENTS)
01:HLT HEAD LOAD TIME IN 2 ms INCREMENTS (BIT #7 IS 3.5" FLAG)
14:GPL GAP4B (WRITE GAP AFTER DATA)
12:MAX SECTORS/TRACK FOR DOUBLE DENSITY
1C:MAX SECTORS/TRACK FOR HIGH DENSITY

THE HI-LITED HEX BYTES ON THE LEFT ARE THE PARAMETERS FOR THE SELECTED DRIVE. YOU MAY CHANGE THEM BY MOVING THE CURSOR TO THE POSITION AND ENTERING THE NEW HEX DATA. THE NEW DATA WILL BE PLACED IN MEMORY. YOU MAY NOW SELECT OTHER DRIVES OR C/R WILL DISPLAY THE COMPLETE TABLE.

If you press the RETURN key instead of selecting a drive, the display of all 10 drives will be put on the screen. Two other hex addresses are also displayed - the track buffer address for the 1772 disk controller and the track buffer address for the 37C65 disk controller. Both of these addresses are set just below 1/2 megabyte which is the minimum memory you can have. If you have more memory and want to move these buffers enter the new hex addresses and enter a control R.

Control X will return to REX without creating a DRIVE(X) file. Control S will ask for the character to name DRIVE and then save the new DRIVE(X) file on the Work drive.

Assuming that you selected the character "A", if you type (#).DRIVEA, the new data will be put in the DIT. Control P will print a copy of the DIT if your printer is active.

If you have more than 10 drive devices (I have 16) you can use DRIVE(X) routines to move these drives on line or off line as you see fit. You might have a special disk which has a special DRIVE(X) routine to access a hidden partition on your hard drive that no one else knows about.

The DIT that is supplied with REXK5 is as follows:

```
LD  TP  B2  B3  B4  B5  B6  B7  REMARKS

00  00  00  03  81  10  12  22  3.5" HD DRIVE PHY #0, LOG #0
FF  00  01  03  01  14  12  1C  5.25" HD DRIVE PHY #1, LOG #1
FF  A0  22  88  01  98  00  00  HARD DRIVE STARTING @ CYL #0
FF  A0  22  88  00  88  00  32  HARD DRIVE STARTING @ CYL #50
FF  A0  22  88  00  88  00  82  HARD DRIVE STARTING @ CYL #130
FF  A0  22  88  00  88  00  D2  HARD DRIVE STARTING @ CYL #210
FF  A0  22  88  00  88  01  22  HARD DRIVE STARTING @ CYL #290
07  02  00  10  00  00  00  10  RAM DRIVE  $100000 - $2FFFFF
FF  A0  22  99  01  98  01  32  HARD DRIVE STARTING @ CYL #306
FF  02  00  20  00  00  00  20  RAM DRIVE  $200000 - $3FFFFF

37C65 TRACK BUFFER ADDRESS $7BC00 17,408 BYTES
1772 TRACK BUFFER ADDRESS $00000 0 BYTES
```

The DIT that is supplied with REXK4 is as follows:

```
LD  TP  B2  B3  B4  B5  B6  B7  REMARKS

00  00  00  03  01  14  12  1C  5.25" HD DRIVE PHY #0, LOG #0
FF  00  01  03  81  10  12  22  3.5" HD DRIVE PHY #1, LOG #1
FF  01  07  20  20  FF  0A  12  5.25/3.5" 720k DRIVE P0,L2
FF  80  22  88  00  88  00  00  HARD DRIVE STARTING @ CYL #0
FF  80  22  88  00  88  00  64  HARD DRIVE STARTING @ CYL #100
FF  80  22  88  00  88  01  0E  HARD DRIVE STARTING @ CYL #270
FF  80  22  88  00  88  01  B8  HARD DRIVE STARTING @ CYL #440
00  02  00  01  00  00  00  06  RAM DRIVE FROM $10000 - $6FFFF
FF  01  07  21  21  FF  0A  12  5.25/3.5" 720K DRIVE P1,L8
FF  01  07  22  22  FF  0A  12  5.25/3.5" 720K DRIVE P2,L9

37C65 TRACK BUFFER ADDRESS $7D800 10,240 BYTES
1772 TRACK BUFFER ADDRESS $7C000 6,144 BYTES
```

The detailed information about the hard drive will be in the HFORMAT and IFORMAT Utilities and the RAM drive info will be in the TURBO utility.

DUMP

HEX AND ASCII DUMP OF A FILE

Syntax:DUMP<,filespec>

The default extension in .BNY. A display of each sector of the file will be displayed on the screen in lines of 16 hex characters and on the right their Ascii equivalents.

DUP

WILL LIST FILES IN FIRST DIRECTORY THAT ARE NOT IN SECOND DIRECTORY.

Syntax:DUP,<drive>,<drive>

ECHO

WILL ECHO A STRING TO THE OUTPUT DEVICE

Syntax:ECHO,<string>

String is a list of ascii characters terminated by a c/r or EOL character. For example, ECHO,Hello World(c/r) would print on the screen "Hello World".

EDD

MINI EDITOR

Syntax:EDD,<filespec>[,<output filespec>]

The default extension is .TXT, the default drive is the working drive. EDD will display a message and then load the requested file. If that file does not exist EDD will create a new file with that name, clear the screen and display a c/r at the home position and then the cursor. The following commands are available:

CTL "I"	: Insert text at cursor to limit of buffer
CTL INS	: Insert one line of text
CTL DEL	: Delete line containing the cursor
INS	: Insert 1 space at cursor position
DEL	: Delete 1 character at cursor position
CTL "E"	: Enter edit mode (Overlay)
CTL "Q"	: Return to DOS saving the text
"?"	: Display command instructions
ESC	: Stop and return to command mode
CTL "X"	: Return to DOS without save
LF ARROW	: Will move cursor left 1 position
DN ARROW	: Will move cursor down 1 line
RT ARROW	: Will move cursor right 1 position
UP ARROW	: Will move cursor up 1 line

EJ

WILL EJECT THE PAPER ON PRINTER #1

Syntax:EJ

EPBURN

COMMUNICATIONS PROGRAM TO WORK THE BP MICROSYSTEMS EP-1 EPROM BURNER.

Syntax:EPBURN

The EP-1 must be connected to COM3.

EXAMINE

A FULL FEATURED DISK REPAIR UTILITY

Syntax:EXAMINE,<drive>

Examine will read the SIR and ask if the Track & Sector values are acceptable. The prompt "COMMAND" is issued and the following commands are available:

R,<sector address>	Read a Sector
W,<sector address>	Write a Sector
D,<sector address>	Read & display a sector
C,<sector address>	Read & display to EOF
M,<byte number>	Modify sector buffer
F,<filespec>	Read first sector of a file
B,<filespec>	Build link table for a file
T,<addr>,addr>,<count>	Move data in memory
S	Return to REXDOS

The sector address is in the form of TTSS where TT is the Track and SS is the sector in hexadecimal format. Several other modifiers are available to help trace files:

+	Will get the next physical sector.
-	Will get the previous physical sector
N	Will get the next logical sector
P	Will get the previous logical sector
=	Will keep the current sector

EXEC

PROCESS A LIST OF COMMANDS

Syntax:EXEC,<filespec>

The default extension is .TXT. Use EDD or EDDI to create a file containing several commands and EXEC will run the commands just as if you have typed them from the keyboard. For example:

Prepare a file named ALLCAT.TXT like this on the Work drive;

```
ECHO,All cat
ACAT,0
ACAT,1
ACAT,7
```

Enter EXEC,ALLCAT

The screen should print "All cat" and then list all the files in each drive separately in alphabetical order.

FORMAT

FORMAT AN HD DISK, 1.2 OR 1.4 MEGABYTES

Syntax:FORMAT,<drive>

If the drive # is missing or is illegal (?) a help message will be displayed. You are asked for a "Disk Name", enter up to 8 characters. Next for a Volume number if you don't enter a Volume # a random number will be assigned. If an HD or Quad Disk is not installed at the logical drive # entered you will be informed. Now the Physical drive # will be displayed on the screen and you must verify that it is correct.

The disk will be formatted with 80 tracks. Then it will be verified to see that all the data is correct. If any byte in a track is bad then that track is unlinked from the free chain of sectors and will not cause any trouble. The disk name extension will be .QDF

In order for a disk to be properly formatted, the correct information must be in the DIT. Correct entries for the HD disk are:

```
LD  TP  B2  B3  B4  B5  B6  B7  REMARKS

00  00  01  03  81  10  12  22  3.50" HD DRIVE PHY #0, LOG #0
01  00  00  03  01  14  12  1C  5.25" HD DRIVE PHY #1, LOG #1
```

FORMATDD (REXK4 only)

FORMAT A 720K DOUBLE DENSITY DISK IN AN HD TYPE DRIVE

Syntax:FORMATDD,<drive>

If the drive # is missing or is illegal (?) a help message will be displayed. You are asked for a "Disk Name", enter up to 8 characters. Next for a Volume number if you don't enter a Volume # a random number will be assigned. If an HD drive is not installed at the logical drive # entered you will be informed. Now the Physical drive # will be displayed on the screen and you must verify that it is correct.

The disk will be formatted with 80 tracks, 720K, 18 sectors per track. Then it will be verified to see that all the data is correct. If any byte in a track is bad then that track is unlinked from the free chain of sectors and will not cause any trouble. The disk name extension will be .DDF

HARD (REXK4 only)

Disk driver for the ST-225,251 hard disk or similar drives. For systems that boot from a floppy disk, HARD must be installed before any access to the hard disk. For systems that boot from the hard disk, HARD must be appended to REXK4H.COM and the combined file linked on the hard disk at cylinder #0 and sector 1. Obviously the hard disk must be working from the floppy disk before you can prepare to boot from the hard disk. The hard disk must be formatted with HFORMAT before it can be used.

Example:APPEND 0.REXK4H.COM,0.HARD.COM,0.REXDOS.SYS

HFORMAT (REXK4 only)

Format a hard disk partition.

Syntax:HFORMAT,<drive #>

The partition on the hard disk will be formatted according to the contents of the DIT for the logical drive # specified. If the logical drive is not a hard drive the program will be aborted.

Setting up a hard disk is quite a problem. Rex can only address 16 megabytes of data so in order to use all of a hard drive you must partition the disk into 2 or more drives. A typical hard drive (ST-225) has 34 sectors on each track and 4 tracks per cylinder (4 heads). The maximum number of cylinders that can be addressed is 255 which only allows about 8.8 megs per partition. You could increase the size of the partition by having more Logical Sectors per Track. This should be a multiple of 34 ($22). The current value is 136 ($88) and the max value could be 238 ($EE).

You must decide how many partitions and how many cylinders you want for each partition. Since H0, the first partition, will be used for the boot and probably be the system drive it doesn't have to be very large. The setup included on the System disk sets H0 to 100 cylinders, H2 & H3 to 170 cylinders and H4 to the rest of the drive, 174 cylinders. The max cylinder number is 615.

You must now create a DRIVE(X) - let's call it DRIVEH - to use when you are using the hard drive. I suggest that you use H0 as L0, Q1 as L1, Q0 as L2, H1 as L4, H2 as L5, H3 as L6 and RAM as L7. The others are not installed ($FF). Type DRIVEH to install the new DIT and then enter 2.SYSTEM,2. This will make drive #2 the system drive and it will contain your system disk.

Go ahead and enter HFORMAT,0. It will take some time to do the low level formatting and then link and test all the sectors. If some sectors are bad don't get upset as most hard drives do have some bad sectors. REX will unlink the bad sectors and you will never know about them again. If you have not installed HARD do so at this time. Now you may CAT,0 and see how many sectors you have. You might also want to COPY 2,0 to get all the utilities over on the hard drive. If you set SYSTEM,0 you will be surprised how much faster everything runs. If everything is successful go ahead and format the other hard drives.

If you need to reserve space for another operating system on the hard disk, such as OS-9, you will need to set up a dummy hard partition with the offset of the reserved area. HFORMAT will search the DIT for any offset greater than its starting cylinder and use the smallest offset as the partition limit. Of course do not format this dummy partition.

To boot the system put a system disk in L0 and enter "V". Now enter HARD and then DRIVEH. Now go read about UPSTART which will do that for you.

HECHO

WILL ECHO TO THE OUTPUT DEVICE THE HEX CHARACTERS THAT FOLLOW

Syntax:HECHO,<hex string>

Hex string is a list of hexadecimal characters.
For example, HECHO,D,A,48,65,6C,6C,6F,7 would output the c/r and l/f then print "Hello" and ring the bell.

HIDE

Disk driver for the ST-3120 IDE hard disk or similar drives. For systems that boot from a floppy disk, HIDE must be installed before any access to the hard disk. For systems that boot from the hard disk, HIDE must be appended to REXK?I.COM and the combined file linked on the hard disk at cylinder #0 and sector 1. Obviously the hard disk must be working from the floppy disk before you can prepare to boot from the hard disk. The hard disk must be formatted with IFORMAT before it can be used.

Example:APPEND 0.REXK?I.COM,0.HIDE.COM,0.REXDOS.SYS

To boot from a hard drive at a partition other than cylinder #0, or if the DIT parameters are different from the REXK?I DIT, the DIT for REXDOS must be overlaid with the current table. Get the hard drive working from a floppy drive with all your partitions etc. Now save a copy of the DIT with 'SAVE 1.DITI.BNY,$3B00,$3B60'. To create the REXDOS for the hard drive enter;

'APPEND 1.REXK?I.COM,1.DITI.BNY,1.HIDE.COM,1.REXDOS.SYS'

Copy REXDOS.SYS to drive 0, LINK it and then use LINKIDE to tell the boot where to find it.

To boot the system from a floppy drive, put a system disk in L0 and enter "V". Now enter HIDE and then DRIVEI. Now go read about UPSTART which will do that for you.

IFORMAT

Format an IDE hard disk partition.

Syntax:IFORMAT,<drive #>

The partition on the hard disk will be formatted according to the contents of the DIT for the logical drive # specified. If the logical drive is not a hard drive the program will be aborted.

Setting up a hard disk is quite a problem. Rex can only address 16 megabytes of data so in order to use all of a hard drive you must partition the disk into 2 or more drives. A typical IDE drive (ST-3120) has 34 sectors on each track and 12 tracks per cylinder (12 heads). This works out to be 408 Sectors per Cylinder ($198). The maximum number of logical tracks that can be addressed is 255. You could increase the size of the partition by having more Logical Sectors per track. This should be a multiple of 34 ($22). A value of 136 ($88) will give an 8.8 megabyte partition and the value of 238 ($EE) about a 15 megabyte partition. The max value is 254 ($FE), it must be even as IDE sectors are 512 bytes and ours are 256.

You must decide how many partitions and how many cylinders you want for each partition. Since H0, the first partition, will be used for the boot and probably be the system drive it doesn't have to be very large. Two sample DITs are shown below and included on the System disk as DRIVEJ and DRIVEK; they set H0 to 50 cylinders, about 5 megs. H1 thru H6 to 148 cylinders and H7 to the rest of the drive, 86 cylinders. The max cylinder number is 1024.

You now use DRIVESET to create a DRIVE(X), let's call it DRIVEI, to use when you are using the hard drive. I suggest that you use H0 as L0, Q1 as L1, H1 as L2, H2 as L3, H3 as L4, H4 as L5, H5 as L6, RAM as L7, H6 as L8 & H7 as L9. The entries under TP(type) are A0 for Master Drive and B0 for Slave drive. Type DRIVEI to install the new DIT and then enter 1.SYSTEM,1. This will make drive #1 the system drive and it will contain your system disk.

Go ahead and enter IFORMAT,0. You will be asked for the last cylinder #, the default is 1024, but you may limit the format to any cylinder. If you enter a C/R the last cylinder will be calculated from the DIT. The starting and ending cylinder numbers will now be displayed and you may abort if necessary. It will take some time to do the high level formatting and then link and test all the sectors. If some sectors are bad don't get upset as most hard drives do have some bad sectors. REX will unlink the bad sectors and you will never know about them again. If you have not installed HIDE do so at this time. Now you may CAT,0 and see how many sectors you have. You might also want to COPY 1,0 to get all the utilities over on the hard drive. If you set SYSTEM,0 you will be surprised how much faster everything runs. If everything is successful go ahead and format the other hard drive partitions.

To reserve space for another operating system on the hard disk, such as OS-9 or MSDOS, which uses cylinder #0. sectors 1 & 2 for its boot loader, start the REX partitions at a higher cylinder to leave space for the other operating systems. LINK REXDOS as usual on H0 and then use LINKIDE a program that will put the DIT of the selected drive # at location $F4 of sector 2, Cylinder 0

SAMPLE DIT (DRIVEJ) with 3.5", 1.4 meg drive at physical #0

```
LD  TP  B2  B3  B4  B5  B6  B7  REMARKS

00  A0  22  88  01  98  00  00  IDE DRIVE STARTING @ CYL #0   H0
01  00  00  03  81  10  12  22  3.5" HD DRIVE PHY #0,         Q0
02  A0  22  EE  01  98  00  32  IDE DRIVE STARTING @ CYL #50  H1
03  A0  22  EE  01  98  00  C6  IDE DRIVE STARTING @ CYL #198 H2
04  A0  22  EE  01  98  01  5A  IDE DRIVE STARTING @ CYL #346 H3
05  A0  22  EE  01  98  01  EE  IDE DRIVE STARTING @ CYL #494 H4
06  A0  22  EE  01  98  02  82  IDE DRIVE STARTING @ CYL #642 H5
07  02  00  08  00  00  00  08  RAM DRIVE FROM $80000 - $FFFFF
08  A0  22  EE  01  98  03  16  IDE DRIVE STARTING @ CYL #790 H6
09  A0  22  88  01  98  03  AA  IDE DRIVE STARTING @ CYL #938 H7

37C65 TRACK BUFFER ADDRESS $7BC00 17,408 BYTES
1772 TRACK BUFFER ADDRESS $00000 0 BYTES
```

SAMPLE DIT (DRIVEK) with 5.25", 1.1 meg drive at physical #0

```
LD  TP  B2  B3  B4  B5  B6  B7  REMARKS

00  A0  22  88  01  98  00  00  IDE DRIVE STARTING @ CYL #0   H0
01  01  00  03  01  14  12  1C  5.25" HD DRIVE PHY #0,        Q0
02  A0  22  EE  01  98  00  32  IDE DRIVE STARTING @ CYL #50  H1
03  A0  22  EE  01  98  00  C6  IDE DRIVE STARTING @ CYL #198 H2
04  A0  22  EE  01  98  01  5A  IDE DRIVE STARTING @ CYL #346 H3
05  A0  22  EE  01  98  01  EE  IDE DRIVE STARTING @ CYL #494 H4
06  A0  22  EE  01  98  02  82  IDE DRIVE STARTING @ CYL #642 H5
07  02  00  08  00  00  00  08  RAM DRIVE FROM $80000 - $FFFFF
08  A0  22  EE  01  98  03  16  IDE DRIVE STARTING @ CYL #790 H6
09  A0  22  88  01  98  03  AA  IDE DRIVE STARTING @ CYL #938 H7

37C65 TRACK BUFFER ADDRESS $7BC00 17,408 BYTES
1772 TRACK BUFFER ADDRESS $00000 0 BYTES
```

In case your IDE drive has more capacity than you can get in 1 DIT, you should split the partitions into 2 or more DITs. It is easy to have several DITs to use for different

programming functions. There is no need to have access to your source code when you are doing business tasks. I would suggest that you keep the original DITs that were used to format the IDE drive in a safe place in case you have to reformat some time. Ron Anderson has a set of routines that let you change partitions just as you change directories in MSDOS (CD) and he will be writing about them soon.

KEYCHEK

WILL DISPLAY IN HEX AND ASCII THE ACTUAL CHARACTER RETURNED FROM A KEY PRESS.

Syntax:KEYCHEK

Control "C" is the exit character and as a result it can't be checked. It is an $03. Don't try the PAUSE or Control NUMLOCK on the PC keyboard or the Control Break on a terminal keyboard. If you are writing a program and want to use the Editing or Function keys, this routine will tell you what character will be returned in D0 after a call to Trap #1.

KRACK

DISASSEMBLER FOR MEMORY OR KEYBOARD ENTRY

Syntax:KRACK

A menu will be displayed and you may choose your options.

LINK

LINK TRACK 0 SECTOR 1 TO THE FILE

Syntax:LINK,<filename>

In order to load the operating system (REXDOS), its address on the disk must be put in Track 0, sector 1, so the loader knows where to start. For example: LINK,0.REXDOS.COM. The default extension is .SYS. Any file can be linked so that other operating systems or programs can be loaded. The program to be loaded must be in REX binary format.

LINKHARD (REXK4 only)

Syntax:LINKHARD,<drive #>

This routine is used when the REX partition is not located at an offset of $00. The offset to the REX partition is put at Cylinder #0, Sector #2, Byte $F4.

LINKIDE

Syntax:LINKIDE,<drive #>

This routine is used when the REX partition is not located at an offset of $00. The offset to the REX partition is put at Cylinder #0, Sector #2, Byte $F4.

LIST

SEND A FILE TO THE OUTPUT DEVICE

Syntax:LIST,<filespec>

The default extension is .TXT. You may optionally enter a Title to be printed on each page.

MAP

LIST THE BEGINNING AND ENDING ADDRESSES OF EACH SEGMENT OF A BINARY TYPE FILE.

Syntax:MAP,<filespec>

The default extension is .BNY

MEMEND

SERVICE THE MEMORY END POINTER

Syntax:MEMEND[,<address>]

Without any arguments MEMEND will display the current Memory End Address in REX. I put a Hex Address after the comma; that address will now become the last available address for REX or programs that run under REX. For example: MEMEND,3FFFF will permit REX to use 256K of memory, reserving the rest for other purposes.

MAKDISK (REXK4 only)

FORMAT A 720K DISK IN THE 1772 TYPE DISK CONTROLLER WITH 1:1 INTERLEAVE

Syntax:MAKDISK,<drive>

Enter the Disk name and then the preset options will be displayed; you may accept them and continue, or reject them and enter your own parameters. A disk formatted this way will run slower on the K2 and older machines.

NEWDISK (REXK4 only)

FORMAT A FLOPPY DISK FOR THE K2 AND OLDER COMPUTERS, USING THE 1772 DC, WITH AN INTERLEAVE OF 3:1, AND OPTIONS FOR DENSITY, SIDES, AND TRACKS

Syntax:NEWDISK

Enter the Disk name and then the preset options will be displayed; you may accept them and continue, or reject them and enter your own parameters. A disk formatted this way will run much slower on the K4.

OLOAD

OFFSET LOADER

Syntax:OLOAD,<filespec>[,<offset>]

The default extension is .BNY. If no offset address is entered then the program will load at its normal address. Offsets can be either negative or positive. For example: OLOAD,0.LIST.COM,1000 would put the LIST utility at $6000.

P10

PRINT AT 10 CHARACTERS PER INCH USING EPSON COMMANDS

P12STAR

PRINT AT 12 CHARACTERS PER INCH USING STAR MICRONICS COMMANDS

PDEL

PROMPTING DELETE UTILITY

Syntax:PDEL,<drive>[,<match list>]

All file names that match will be displayed on the screen with the option to delete or continue.

PE2

PRINT 12 CHARACTERS PER INCH ON PRINTER #2

PN2

PRINT NEAR LETTER QUALITY ON PRINTER #2

PNLQ

PRINT NEAR LETTER QUALITY ON PRINTER #1

PRINT(PRDVR)

PRINTER DRIVERS

Syntax:PRINT(XX) (PRDVR(XX))

The (XX) are two characters which determine the ports that printer #1 and printer #2 will use. There are 2 parallel ports on the computer, PORTA and PORTB, and 4 Serial ports, COM1, COM2, COM3, COM4. The addresses for these ports are in the memory map supplied with MONK. The printer driver that comes installed in REX is PRINTA2, meaning that printer #1 outputs on PORTA, a parallel interface, and printer #2 outputs on COM2, a serial interface. For example: PRINT43 would make printer #1 go to COM4 and printer #2 go to COM3.

PS

PRINT SMALL (16 CPI) ON PRINTER #1 (EPSON COMMANDS)

PS2

PRINT SMALL (16 CPI) ON PRINTER #2 (EPSON COMMANDS)

RAWKEY

DISPLAY PC KEYBOARD KEYPRESS DATA

Syntax:RAWKEY

DISPLAY ON A TERMINAL THE ACTUAL DATA RETURNED BY KEYPRESSES ON THE PC KEYBOARD.

REDATE

CHANGE FILE CREATION DATE

Syntax:REDATE,<filespec>[,MM,DD,YY]

If the date is omitted or bad then the current system date will be used.

RENAME

CHANGE THE NAME AND EXTENSION OF A FILE

Syntax:RENAME,<filespec>,<new filespec>

The default extension is .TXT and the default drive is work drive. If no extension is given for the new filespec the old filespec extension will be used. A drive # in the new filespec is ignored.

REVERSEC

SET REVERSE VIDEO COLOR

Syntax:REVERSEC,[color #]

Default is 71 white/blue characters. 24=green/red, 42=red/green, 34=Blue/Red, 43=Red/Blue, 14=Blue/Red, 41=Red/Blue, other combinations are available.

SAVE

SAVE MEMORY AS A REX BINARY FILE OR AS A MEMORY IMAGE FILE

Syntax:SAVE,<filespec>,<beg addr>,<end addr>[,+][tran addr]

The default extension is .BNY, unless the (+) option is selected, when an extension of .MEM is forced. The transfer address is ignored if the (+) option is selected and if it is not included in a binary save the file will be saved without a transfer address. If the filespec exists on the destination drive you will be asked if you want to delete it.

SAVETEXT

SAVE A TEXT FILE TO DISK

Syntax:SAVETEXT,<filespec>,<beg addr>,<end addr>

The data will be saved as a text file with space compression.

SCAN

A MINI SHELL PROGRAM FOR REX

Syntax:SCAN,<drive>

The default drive is the work drive. The disk directory is displayed in a mask and the following commands are active.

C	Will copy the Entry under the Cursor from the Work Drive to the Designated Drive
D	Will Delete the File under the Cursor and remove the Entry from the Catalog
ESC	Will Terminate Viewing and Return to the Catalog
F	Will Execute a Rex Command.
/	Will enter the File under the Cursor in the command line
H	Will display these messages
I	Will Install a Binary type File under the Cursor to the Designated Drive with a .COM extension
V	Will View the File under the Cursor
X	Will Exit to Rex

SET(XX)

INSTALL MODES TO THE VGA CONTROLLER CARD

Syntax:SETXX

The following routines were developed to test the VGA Mode selections of Trap #12.

SET1	Set Video Mode 1 40x25 Text
SET3	Set Video Mode 3 80x25 Text
SETE	Set Video Mode E 640x200 Graphics
SET10	Set Video Mode 10 640x350 Graphics
SET12	Set Video Mode 12 640x480 Graphics
SET13	Set Video Mode 13 320x200 Graphics
SET22	Set Video Mode 22 132x44 Text
SET23	Set Video Mode 23 132x25 Text
SET24	Set Video Mode 24 132x28 Text
SET26	Set Video Mode 26 80x60 Text
SET29	Set Video Mode 29 800x600 Graphics
SET2A	Set Video Mode 2A 100x40 Text
SET2D	Set Video Mode 2D 640x350 Graphics
SET2E	Set Video Mode 2E 640x480 Graphics
SET2F	Set Video Mode 2F 640x400 Graphics
SET30	Set Video Mode 30 800x600 Graphics
SET37	Set Video Mode 37 1024x768 Graphics
SET38	Set Video Mode 38 1024x768 Graphics
SET53	Set Video Mode 53 80x50 Text

SETPC

PUT THE PC KEYBOARD AND VGA IN CONTROL

Syntax:SETPC

The Trap #1 and #2 vectors will now point to the PC keyboard and VGA screen routines.

SETTERM

PUT THE TERMINAL IN CONTROL

Syntax:SETTERM

The Trap #1 and #2 vectors will now point to the Terminal routines.

SETIME

SET THE CLOCK AND CALENDAR CHIP

Syntax:SETIME

A menu to start or stop the clock oscillator is displayed. If you are going to store the computer for a while or not use it. Turning the oscillator off will extend the battery life. You will be requested to enter the minutes, hours, am or pm, day of the week, day of the month, month, and year. When you press a character to start the clock it will start with the seconds at 00. Next the correction factor will be displayed and the opportunity to change it. Each correction will change 5.35 seconds per month.

SLIDE

DISPLAY PICTURE FILES

Syntax:SLIDE,<Filespec>,[Filespec],.....[Filespec]

The default extension is .PIC. The maximum number of files to be displayed is limited by the LINE BUFFER size of 128 bytes. The ESC character will return to DOS and any other character will display the next picture. Use EXEC to create a larger slide show.

SPOOLA

Syntax:SPOOLA

SPOOLA is a printer driver for parallel port A which stores the characters to be printed in memory and returns to REX. When the printer is ready to accept a character it interrupts the computer for a few microseconds to get a character to print. This process continues until the document is finished. If you go to MONK or any other program which sets the interrupt mask to level 3 or higher the printing will stop until the mask is cleared. The memory from $80000 to $FFFFF is reserved for Spooling. The SPOOLA loads below $5000. Multiple documents can be sent to the spooler; however, if the document doesn't do a form feed at the end you should do an EJ command between documents to start the next document at the top of a page. Caution, the RAM disk(TURBO) may conflict with SPOOLA.

SYSTEM

SET THE SYSTEM DRIVE #

Syntax:SYSTEM[,<drive>]

Without any arguments SYSTEM will display the current system drive #.

TCOPY

A FAST COPY ROUTINE

Syntax:TCOPY,<source drive>,<dest drive>[,matchlist]

TEXTC

CHANGE THE TEXT COLOR

Syntax:TEXTC,[color #]

The default is 0E=Black with Yellow characters. 1E=Blue/yellow, 1F=Blue/White, 4E=Red/Yellow, 0F=Black/White, 5F=Magenta/White, 6A=Brown/Green, 70=White/Black, 74=White/Red, other combinations are possible.

TFRATE

MEASURE THE TRANSFER RATE OF DISK READS

Syntax:TFRATE,<filespec>

The selected file will be read and the elapsed time, byte count, and transfer rate will be displayed.

TIME

DISPLAY THE TIME AND DATE

Syntax:TIME

The time and date will be displayed on the screen.

TIMER

MEASURE ELAPSED TIME

Syntax:TIMER

The ESC key will return to DOS. Any other key will start and stop the timer. The elapsed time will be displayed in Microseconds.

TURBO

INSTALL A RAM DISK

Syntax:TURBO,<drive #>

If a drive # is not entered then the default drive #7 will be installed. If the drive # specified is not a RAM drive you will be informed. If the RAM disk has already been installed you will be informed and the current contents will be retained.

Use DRIVESET to define the location of a RAM disk. Byte #1 must be $02 to be RAM drive. Bytes 2-5 are the starting address of the RAM disk and bytes 6-7 are the number of 64K pages of memory to use.

e.g.

```
LD  TP  B2  B3  B4  B5  B6  B7  REMARKS

07  02  00  01  00  00  00  06  390K AT $10000 AS L7
08  02  00  10  00  00  01  00  1 MEG AT $100000 AS L8
```

The max size of a RAM disk is 4 megabytes.

VIEW

VIEW A FILE

Syntax:VIEW,<filespec>

The default extension is .TXT, the default drive is the work drive. The selected file will be displayed on the screen. The down arrow will move forward one line and the up arrow will move the display backwards 1 line. The ESC will return to REX.

WORK

SET THE WORK DRIVE #

Syntax:WORK[,<drive>]

Without an argument WORK will display the current work drive #.

XTURBO

UNINSTALL A TURBO DRIVE

Syntax:XTURBO,<drive #>

The drive # must be included. You will be asked if you want to delete the selected drive; a no answer will return to DOS.

YEAR

DISPLAY OR CHANGE THE REX YEAR BYTE

Syntax:YEAR[,YY]

YEAR without any parameters will display the current REX year byte. Followed by a hex number the REX year byte will be changed.
For example YEAR,89 would set the year byte to 1989.

ZAP

DELETE ALL FILES ON DISK

Syntax:ZAP[,drive list][,match list]

For example, ZAP,1,.BAK would delete all files on drive #1 with a .BAK extension.

REX MEMORY MAP

0400 - 0FFF	HARD DISK DRIVER
1000 - 1640	USER FCB'S
1640 - 20FF	REX STACK
2100 - 2240	SYSTEM FCB
2280 - 22FF	LINE BUFFER
2400 - 4FFF	REXDOS
5000 - 5FFF	UTILITY COMMAND SPACE
6000 - 6FFFF	APPLICATION RAM
70000 - 7BBFF	SUBCAT OR OTHER SHELLS
7BC00 - 7FFFF	TRACK BUFFERS
80000 - 100000	SPOOLER RAM
100000 - 3FFFFF	USER RAM

REX ERROR NUMBERS

ERROR #	Interpretation
1	ILLEGAL FUNCTION CODE
2	FILE IS IN USE
3	FILE EXISTS
4	FILE NOT FOUND
5	DIRECTORY IS FULL
7	ALL DISK SPACE HAS BEEN USED
8	READ PAST END OF FILE
9	DISK READ ERROR
10	DISK WRITE ERROR
11	DISK IS WRITE PROTECTED
13	ILLEGAL FILE CONTROL BLOCK
14	ILLEGAL DISK ADDRESS
15	ILLEGAL DRIVE NUMBER
16	DRIVE NOT READY
18	SYSTEM FILE STATUS ERROR
21	ILLEGAL FILE SPECIFICATION
22	SYSTEM FILE CLOSE ERROR
26	COMMAND SYNTAX ERROR
29	SEEK ERROR