

TABLE OF CONTENTS

	<u>QUICK</u>	<u>DETAIL</u>
FM - Overview.....	3	
DRIVE NUMBER USAGE:		
In Display mode .....	4	14
In Kill mode .....	4	15
In Move mode .....	4	16
In the Remove Mode .....	5	17
USING PARTSPECS:.....	5	17
WILDCARD CHARACTERS: .....	6	21
PARAMETERS:		
EXISTENCE PARAMETERS:		
ABS .....	7	
CK (Check NEW with Move or Remove) .....	7	
N (New) .....	7	
See individual Display, Kill, Move or Remove sections for detailed descriptions of ABS, CK, and NEW		
DISPLAY PARAMETERS:		
A (Allocation) .....	7	24
G (Gran).....	7	24
K (1024 byte		
O (Sort) .....	8	25
P (Printer) .....	8	26
Q (Query) .....	8	26
ATTRIBUTE PARAMETERS:		
I (Invisible) .....	9	27
S (System) .....	9	27
V (Visible) .....	9	27
CM (Clear MOD) .....	9	27
SM (Set MOD) .....	9	28
FB (Use Flag Bit) .....	9	28
CFB (Clear Flag Bit) .....	9	28
SFB (Set Flag Bit) .....	10	28
PR (Use Protection Level) .....	10	29
DATE PARAMETERS:		
D (Date) .....	10	29
SD (Set Date) .....	10	30
T (Today) .....	10	30
MODIFICATION PARAMETERS:		
M (Modified) .....	11	31
U (Unmodified) .....	11	31
SIZE OF FILE PARAMETERS:		
L (Limit) .....	11	31
Z (Size) .....	12	31

	<u>QUICK</u>	<u>DETAIL</u>
JCL OPERATION PARAMETERS:		
JCL .....	12	32
ABORT .....	12	32
FILE OUTPUT PARAMETERS:		
FI (File) .....	12	32
STR (Substitution string) .....	12	33
E (Enter character) .....	13	33
FINAL MESSAGE DISPLAY:		
In the Display mode .....		34
In the Move mode .....		34
In the Kill mode .....		34
In the Remove mode .....		34
ERROR HANDLING:		
Disk I/O errors.....		34
HELP SCREEN DISPLAY.....		36

## FM UTILITY PROGRAM

FM stands for File Manager. It is a utility program designed to facilitate specific manipulation of files. Four modes are available - Display, Kill, Move, and Remove, with the Display mode being the default. The basic syntax and available parameters for the command are:

```
FM partspec:d (parm,parm,...)

FM partspec:d1 :d2 (parm,parm,...)

FM partspec:d (KILL,parm,parm,...)
FM partspec:d1 :d2 (KILL,parm,parm,...)

FM partspec:d1 :d2 (MOVE,parm,parm,...)
FM partspec:d1 :d2 :d3 (MOVE,parm,parm,...)

FM partspec:d1 :d2 (REMOVE,parm,parm,...)
FM partspec:d1 :d2 :d3 (REMOVE,parm,parm,...)
```

The first three modes correspond with the commands DIR, PURGE, and the BACKUP by Files utility. The Remove mode is a combination of BACKUP and PURGE, with the files being moved from a source to a destination drive, and then being removed from the source drive. One of the main features of FM is that file operations can involve more than the usual number of drives. This provides comparison opportunities that can simplify the maintenance of diskette sets. The partspecing abilities of FM include three wildcard characters as well as the ability to specify a separate filename and extension for comparison purposes. The parameters include those previously available with BACKUP, such as modified, visibility status, file dating, etc. New parameters deal with recent dates, unmodified files, a way to set or clear mod flags without actually moving the files, and more. Due to the increase in popularity of large volume disk drives, several special parameters have been added to facilitate moving files from these larger drives onto smaller volume diskettes. Lastly, the speed of moving files has been increased by approximately 50% over the normal BACKUP by files, yet still includes a full read verify.

For reference purposes, the wildcard characters and the parameters are:

### **WILCARDING CHARACTERS:**

```
$   Used as a masking character
*   Used to indicate an Instring function
!   Used to indicate truncation of character checking
```

```
A (Allocation), O (Sort), P (Printer), Q (Query)
GRAN(G), K (1024 byte blocks)
NEW (N), ABS, CK (Check)
MOD (M), U (Unmodified), VIS (V), INV (I), SYS (S)
DATE (D), TODAY (T)
CM (Clear Mod), SM (Set Mod), SD (Set Mod date),
CFB (Clear Flag Bit) SFB (Set Flag Bit) FB (Use Flag)
PR (Use Prot)
SIZE (Z), LIMIT (L)
JCL, ABORT
FILE (FI), STR (Substitution string) ENTER (E)
KILL (KI), MOVE (MV), REMOVE (RMV)
```

As can be seen from the layout of the first command block, FM has four modes (Display, Kill, Move and Remove), and can be considered to have three fields for any mode; the PARTSPEC, the DRIVE NUMBERS, and the PARAMETERS. Since the function of the drive number(s) is very important to FM, it will be explained first. The explanation of partspecs will follow, with the parameters being discussed last.

#### **DRIVE NUMBERS:**

The use of drive numbers depends on the mode used. In the Display and Kill modes, either one or two drive numbers can be used. In the Move or Remove modes, at least two drives must be specified, and three drives can be involved. When multiple drive numbers are used, the first will be always be referred to as the SOURCE DRIVE, and the second as the DESTINATION DRIVE. This will be true regardless of the mode involved. When using a three drive Move command the third drive will be referred to as the COMPARISON DRIVE. Although the following examples all use drives 0 and 1, any drive numbers may be used.

#### **IN THE DISPLAY MODE:**

```
FM :0  
FM :0 :1
```

These two commands show the use of drive numbers in the Display mode. The first command means "Show me all the files on drive 0 (the source drive) that match any partspec or parameters I have specified". The second command means almost same thing, except that using the second drive number says "Display the files on drive 0 based on a comparison of the files on drive 1". The important thing to remember is that the displayed files will always be files that are on drive 0, the source drive. Therefore, the Display mode can be used to "preview" the action of a Kill or Move command, and can assure that only the desired files will be acted upon.

#### **IN THE KILL MODE:**

```
FM :0 (KILL)  
FM :0 :1 (KILL)
```

These two commands show the use of drive numbers in the Kill mode. The first command means "Kill all the files on drive 0 (the source drive) that match any partspec or parameters I have specified". The second command means almost same thing, except that using the second drive number says "Kill the files on drive 0 based on a comparison of the files on drive 1". The important thing to remember is that the Killed files will always be files that are on drive 0, the source drive in this example. No file on drive 1 will be affected. To be assured that proper files will be killed, first use the identical command without the KILL parameter (i.e., the Display mode) to view the files that matched the specifications used.

#### **IN THE MOVE MODE:**

```
FM :0 :1 (MOVE)  
FM :0 :1 :2 (MOVE)
```

The Move mode is a little more complex as far as the use of drive numbers -is concerned. The important thing to remember is that files will always be moved from the first drive to the second, regardless of the use of a third drive. When using a two drive Move command, the second drive, besides being the destination drive, also becomes the comparison drive. Thus the two drive command means "Move files that match the partspec and parameters from drive 0 to drive 1, based on a comparison of drive 1". The three drive command means "Move the files from drive 0 to drive 1 based on a comparison between drives 0 and 2". To be sure the proper files will be moved, use the command without the MOVE parameter to display the matching files. For the first example, the

corresponding Display mode command would be FM :0 :1, as only two drives are involved. To view the matches of the second Move example, the Display mode command FM :0 :2 should be used, as drive 2 rather than drive 1 is the comparison drive.

#### IMPORTANT

The NEW and ABS parameters, as discussed in a later section, are very important in MOVE and REMOVE commands. If in doubt concerning their use, study the detailed discussion section.

#### IN THE REMOVE MODE:

```
FM :0 :1 (REMOVE)
FM :0 :1 :2 (REMOVE)
```

The Remove mode uses drive numbers in the same manner as the Move mode. That is, that files will always be moved from the first drive to the second, regardless of the use of a third drive. When using a two drive Remove command, the second drive, besides being the destination drive, also becomes the comparison drive. Thus the two drive command means "Move files that match the partspec and parameters from drive 0 to drive 1, based on a comparison of drive 1, and then kill the files on drive 0". The three drive command means "Move the files from drive 0 to drive 1 based on a comparison between drives 0 and 2, and then kill the files on drive 0". To be sure the proper files will be moved, use the command without the REMOVE parameter to display the matching files. F or the first example, the corresponding Display mode command would be FM :0 :1, as only two drives are involved. To view the matches of the second Move example, the Display mode command FM :0 :2 should be used, as drive 2 rather than drive 1 is the comparison drive.

#### PARTSPECS:

Partspec stands for Partial File Specification. A file specification is defined as being a filename up to 8 characters long followed by an optional extension of a "/", followed by up to three characters. A partspec is considered to be any or all parts of a file specification. One important point to remember is that passwords are not necessary in any FM command, and should NEVER be used.

#### SPLITTING PARTSPECS:

The normal LDOS use of partspecs allows specifying a filename, an extension, or a filename/ext. The NOT symbol "-" can also be used to show an exclusion partspec such as -filename, -/ext, or -filename/ext. FM supports these standard uses of a partspec and adds to them! To accomplish this, FM allows a comma "," to be used as a separator between the filename and extension. Additionally, a filename may now include a trailing "/", meaning only files with no extensions. The following table lists the different partspec modes:

filename or -filename This type of partspec lets you specify either an inclusion filename or an exclusion filename. Any file whose filename matches the criteria will be a match regardless of the presence or absence of an extension.

filename/ or -filename/ This example is similar to the previous one, except that only files without extensions will be considered.

/ext or -/ext This type of partspec lets you specify an inclusion or exclusion extension. Since all files must have filenames, any file with an extension matching the criteria will be valid.

filename/ext or -filename/ext This example lets you specify an inclusion or exclusion filename as long as the files all match the extension criteria.

{-}filename,{-}/ext The power of having separate filenames and extensions makes itself felt by having the ability to specify an inclusion or exclusion filename, and a separate inclusion or exclusion extension. The braces "{}" mean that the NOTs are optional; the braces themselves should not be included on the command line. Following are the combinations using the separation of filename and extension:

filename,/ext    -filename,/ext    filename,-/ext    -filename,-/ext

#### **WILDCARD CHARACTERS IN PARTSPECS:**

FM allows the use of three wildcard characters whenever a partspec is used. These are \$ (Mask character), \* (Instring character), and ! (Truncate character). More than one of these characters can be used at the same time, and in either the filename and/or extension fields.

\$ - The mask character is used to indicate that any character in a given position will be considered a match. For example, a partspec of \$\$A means "match all files that have an A as the third letter of the filename, regardless of the other characters in the filename." The \$ may also be interspersed in a filename, such as T\$\$60\$1. Using the \$ as a trailing character, such as TE\$\$\$\$, does NOT mean "only those files starting with TE and that contain 5 characters". The normal LDOS partspecing takes a partspec "TE" to mean "all files starting with TE, regardless of any other characters which may follow". To specify filenames of a certain length, see the ! character. The \$ character can be used in the extension part of a partspec as well as in the filename portion.

\* - The instring character is used to find a match of a specified string of characters anywhere in a filename or extension. The string to find may contain the \$ mask character. For example, a partspec of \*AT would find a match in files named PATCH, ATTACK, FORMAT, etc. A partspec such as \*B\$\$ would find files such as BASIC, OLDBUSI, etc. A partspec of /\*C would match all files that had a C anywhere in the extension. As filenames and extensions can be a maximum of 8 and 3 characters long, respectively, the characters following an \* are limited to 7 for a filename and 2 for an extension.

! - The truncate character is used to indicate files that have a filename or extension of a specified length. For example, a partspec of ALF! would match a file named ALF, but not ALFA1, ALFALFA, or AL. The \$ character can also be used with the !. A partspec such as TE\$\$\$\$! would find all files that start with TE and exactly five characters long. A partspec of /T\$! would match all files whose extensions start with a T and are exactly two characters long.

#### **PARAMETERS:**

The parameters of FM can be considered to be grouped into classes by function. Certain parameters deal with the attributes of files, some with dates, and others with size. To allow FM to be controlled by a JCL file, the JCL and ABORT parameters are included. The KILL, MOVE and REMOVE parameters switch FM out of its normal Display mode. One group of parameters deals with the type of display you will get from FM; sorted or unsorted, to the video or printer, and prompt or go nonstop.

#### **FILE EXISTENCE PARAMETERS:**

The NEW and ABS parameters can be used to override the default "Old comparison" feature of FM. NEW is valid in all four modes, while ABS is only valid for the Move and Remove modes. The CK (Check) parameter can be used along with NEW to determine if there is enough room on the destination files when moving files from one drive to another.

## NEW (N)

The NEW parameter can be included in any FM command that uses two or more drives; in one drive commands, NEW is meaningless. The normal default for any FM command regardless of the mode can be considered to be OLD. This means that only those files that exist on both of the drives will be considered. For example, the command "FM :0 :1" will show only those files that are on both drives 0 and 1. To see the files that are on drive 0 but not on drive 1, the command "FM :0 :1 (NEW)" can be used. The NEW parameter is valid in all four modes.

## CK (Check)

The purpose of the CK parameter is to check the amount of free space on the destination disk when the NEW parameter is used along with MOVE or REMOVE. If there is not enough free space to hold the new files, an appropriate error message will be shown, and the operation will abort.

## ABS

The ABS parameter can be used in the Move and Remove modes. The normal functioning of these modes is to only consider files that exist on both drives specified. If ABS is used, FM will act on the files whether or not they exist on the destination (or comparison) drive.

## **DISPLAY PARAMETERS:**

### A (Allocation)

The A parameter is used to display certain information about a file besides the filename and extension. The format of the display will be:

```
|MM/DD/YY| nnnR  
+MM/DD/YY+ nnnR
```

The difference between these two examples is the use of | and + separators surrounding the date field. The date enclosed is the Mod date, and shows the date on which the file was created or last written to. If the | character is used, it means that the file's mod flag is not set. A + character indicates that the file has been modified (the FM parameters SM and CM may be used to adjust a file's Modified condition). The "nnnR" is the file's size in full disk records (256 byte blocks). The GRAN and K parameters may be used to switch from the number of records to the number of grans of a specific size, or to the number of 1024 byte blocks. In the Display mode, the normal screen or printer output will show the files in four across format. The A parameter switches to single line output. The Kill, Move, and Remove modes automatically show the mod date and file size.

### GRAN=nn (G)

The GRAN parameter is used to display the size of a file in grans. Since LDOS supports many different disk types, the number of sectors per gran to use for the calculation may be entered by the user. If the GRAN parameter is used with no value, a 6 sector gran will be assumed. This will match the format of a 5" double density disk. The total grans of all matching files will be shown on the last line of the display.

### K (1024 byte blocks)

The K parameter is used to display the size of a file in K, or 1024 byte blocks. No "partial" K will be shown; any remainder will be rounded up to the next larger value. For example, 4.5 K will show as 5K. The total K of all matching files will be shown on the last line of the display.

### O=ON/OFF (Sort)

Functionally, the O parameter is used to turn OFF the alphabetic sort feature of FM. Normally the Display mode show files sorted alphabetically, regardless of the order in which they are encountered when searching the disk. This is very useful for locating a particular file. On the other hand, the Kill, Move, and Remove modes act on the files in the order they are encountered. Thus, at times it may be desirable to display the files in their unsorted order before using one of those commands. If O is not specified when doing a Display mode FM command, it defaults to on.

### NOTE

Sorting the file specifications requires a certain amount of free memory. If there is not enough memory available, FM will not display the files in alphabetical order. This will be a very unusual occurrence, but can happen if free memory drops below a certain point.

### P (Printer)

Normally, the files acted on are shown on the video display. The P parameter provides a way to send the results to a printer. In the Display mode, the files are normally sent to the printer four across. If the A parameter is used, each file will be printed on a separate line. If the P parameter is used in the Kill, Move, or Remove mode, the file display will also be sent to the printer in single line format.

### Q=ON/OFF (Query)

The Q parameter is used in all four FM modes. With the Kill, Move and Remove modes, it lets the user be queried whether to act on the currently displayed file. For Kill and Remove, the default is ON (query before a file is killed); for Move the default is off (move files without asking). The default may be overridden by specifying the desired state of the Q parameter. In the Display mode, a Q=OFF parameter will keep the screen display from pausing as it normally does every 11 lines, and will display all files non-stop regardless. In the Kill, Move, and Remove modes, the following prompt will appear if Query is ON:

(Y/N/C) ?

Pressing <Y> tells FM to act on the file; pressing either <N> or <ENTER> means to bypass the file without action. pressing <C> will act on the file and turn OFF the Query function. FM will continue nonstop from this point.

### **FILE ATTRIBUTE PARAMETERS:**

The next group of parameters lets the user specify groups of files based on their directory attribute type or modification status, as well as whether or not they exist on a destination or comparison drive. Two parameters, SM (Set Mod flag) and CM (Clear Mod flag) are provided to either set or clear the modification flags on specified files. Three parameters deal with the "Flag bit". This bit is also used by the PDS utility (a product of MISOSYS) to indicate its files. LDOS indicates files with this bit set by displaying an asterisk after their name in a DIRectory display. FB will find all files with the Flag bit set, SFB will set the Flag bit on matching files, and CFB will clear the Flag bit. The PR parameter will find all files matching a specified protection level, 0-7 (as described in the LDOS library command ATTRIB).

#### INV (I)

The INV parameter means act on only those files that are marked in the directory as being INVisible. It can be used in conjunction with the VIS and SYS parameters as required.

#### VIS (V)

The VIS parameter means act on only those files that are visible in a directory display. If neither INVisible nor SYStem are used as parameters, FM will use VIS as a default. To combine directory type parameters, use a command such as (V,I) meaning both visible and invisible, (V,S) meaning visible and system files, or (V,I,S) meaning all file types.

#### SYS (S)

The SYS parameter is used to specify LDOS system files. It can be used along with the VIS and INV parameters to specify file groups as needed. When using the S parameter in the Move or Remove mode, FM will correctly place LDOS system files in their required directory slots. If the SYS0/SYS file is moved, FM will also move the System Information Sectors as described in the LDOS documentation. This means that the state of any sysgened configuration, the state of the power-up date and time prompts, and the default drive configurations will be moved to the destination disk.

#### CM (Clear Mod flag)

The Display mode, using the CM parameter will clear the mod flag from any specified file. In the Move mode, using CM will always clear the mod flag on the source and destination drives. Kill mode ignores this parameter.

#### SM (Set Mod flag)

In the Display mode, using the SM parameter will set the mod flag on every file that matches the specified partspec and/or parameters. In the Move mode, using SM will set the mod flag on both the source and the destination drive. In the Remove mode, the destination mod flag will be set. Since the source file will be killed, the SM parameter does not consider it.

#### FB (Use Flag Bit)

The "Flag" bit is a method of marking a group of files. The LDOS compatible PDS utility program also makes use of this directory marking feature. FM provides the ability to mark desired files in this manner, thereby allowing them to be Displayed, Moved, or Killed as a group. If a file has this bit set, it will appear in a normal LDOS DIR command with an asterisk after its name. The FB parameter can be used to find these files.

#### CFB (Clear Flag Bit)

This parameter provides a means to clear the user "Flag" bit discussed in the FB and SFB parameters. To see files that have this flag bit set, do a DIR command and observe those files that have an asterisk after the filespec.

### SFB (Set Flag Bit)

This parameter provides a means to set the user "Flag" bit for a file. It can be used as a means of marking a group of files to be manipulated in a common manner. To see files that have this flag bit set, do a DIR command and observe those files that have an asterisk after the filespec.

### PR (Use PROTection Level)

In the Display mode, this parameter limits the display to those files that have a matching Protection level. Normally, this protection is linked with the use of a password assigned to a file. The different modes will act on only those files that match the protection level. This protection level may be assigned with the LDOS library command ATTRIB.

## **DATE SPECIFICATION PARAMETERS:**

Three FM parameters deal with a file's date; two of them are used to select files based on a date, while the third can be used to change a file's date. The term "date" when used in reference to a file is the mod date - the date the file was created or last written to.

### DATE (D)

The date parameter is used to specify a single date or a range of dates that will be used to determine if a file should be acted upon. Four forms of the parameter are allowed:

D="MM/DD/YY" Files only with the specified date.

D="M1/D1/Y1-M2/D2/Y2" Files with dates between the first and the second, inclusive.

D="-MM/DD/YY" Files with dates less than or equal to the specified date.

D="MM/DD/YY-" Files with dates greater than or equal to the specified date.

The D parameter may be used in all modes.

### SD (Set Mod date)

The SD parameter is use to set the modification date of specified files. The date can be set either to the current system date or to a user specified date. The format for this parameter is:

SD This command will use the current system date.

SD="MM/DD/YY" This command will use the specified date string.

In the Move or Remove mode, using SD will set the dates on the destination drive files. The files on the source drive will retain their original dates.

### TODAY (T)

One of the most common uses of the date parameter is to deal with files that have been recently modified. In recognition of this, the TODAY, or T, parameter has the ability to define a range of dates relative to the current setting of the LDOS system date. The allowable range is from seven days previous to the current date. The format of the T command is:

T This command means files with today's date.

T="n" where "n" is a number 1 to 7; 1 meaning yesterday and 7 meaning seven days ago.

T="n1-n2" where n1 is greater than n2. This command means "from n1 days ago to n2 days ago".

T="-n" This command means "from seven days ago to n days ago".

T="n-" This command means "from n days ago to today".

The T parameter can be used all modes.

#### **FILE MODIFICATION PARAMETERS:**

The file modification parameters are used to select either files that have been modified or files that have not been. With LDOS, a file is considered to be modified if it has been written to since it was last backed up.

##### MOD (M)

The MOD parameter can be used to specify only those files that have their "Mod" flags set. In normal LDOS operation, a file's mod flag will be set any time a file is written to or changed in any manner. The FM parameters SM (Set Mod) and CM (Clear Mod) can be used to adjust a file's Mod status.

##### U (Unmodified)

This parameter can be used to specify only those files that have not been written to or changed. As with the MOD parameter, the SM and CM parameters can be used to change a file's Mod status.

#### **FILE SIZE PARAMETERS:**

Two parameters, Limit and Size, can be used very effectively in systems that need to move data from large volume drives to smaller ones. While the FM utility does not provide a way to move a single large file onto a series of smaller disks, it does give the user an easy means to bypass these large files in Move or Remove commands, and then see which files were not moved.

##### LIMIT (LM)

The Limit, or LM, parameter is used to specify a maximum file size. The value for the Limit parameter will usually reflect the size of the destination disk to be used in the Move mode. For example, a 49 track, double density, single sided disk has 174K free after formatting. A 35 track, single density, single sided disk has 83K free after formatting. When moving files from a larger to a smaller disk, you can use the Limit parameter to have the FM utility ignore any files that will not fit on a single destination disk. For example:

```
FM :0 :1 (MOVE,LM=174)
```

This command tells FM not to attempt to move files larger than 174K. Once the Move has finished, the Size parameter can be used to see any files that have not been moved.

## SIZE (Z)

The Size, or Z, parameter is used to view files that are larger than a specified length. This parameter can be used in conjunction with the Limit parameter when moving files from a large volume drive to multiple smaller disks. The format is:

SIZE=nnn

The "nnn" is the size in K (1024 byte blocks).

## JCL PARAMETERS:

The JCL (Job Control Language) parameters provide a means to run the FM utility under the control of the LDOS JCL processor. The use of these parameters is necessary due to the automatic error recovery feature of FM. During certain operations, FM will prompt the operator if a disk I/O error has occurred, asking whether to retry the operation, skip the file in question, or abort the operation entirely. This unexpected prompt could cause an active JCL to get "out of sync", and random results could follow. To avoid any problems, the following two parameters can be used.

JCL=Retries

This parameter tells FM that the current operation is being controlled by a JCL file, and also lets the operator specify the number of automatic retries to be attempted in case of an error. If JCL is used with no value, two retries will automatically be done. In either case, if the retries are not successful, the file will be skipped and the FM operation will continue. The one exception to this is the occurrence of a "Disk full" error, which will abort an FM command running under JCL control.

## ABORT

The ABORT parameter can be used to override the automatic retry function that occurs when running under JCL control. If ABORT is specified, the occurrence of any error will abort the FM command, and the JCL processing.

## OUTPUT FILE PARAMETERS:

The output file parameters of FM are ENTER, FILE and STRING. They provide a means of writing any matching filespecs to an ASCII disk file. If desired, the filespecs can be embedded in a larger ASCII string, with the entire string being written to an output file. The main purpose of this feature is to create JCL files that can later be executed to perform a desired function on a group of files.

FILE="filename"

The FILE parameter is used to specify the name of the output file. If no extension is specified, /JCL will be the default. If the FILE parameter is used with no "filename", a file called FMOUT/JCL will be written. If an error condition is encountered when writing the output file, a single error message will be displayed but the FM command in progress will continue. The STRING parameter can be used to specify the characters that will be output to the file. If it is omitted, the only characters written to the output file will be the filespec, including the drive number of the source drive.

STRING="desired output string"

The STRING parameter can be used to specify the line sent to an output file. It allows for insertion of matching filespecs with and without drive number and password, and can embed <ENTER> characters in the string. Each string sent

will automatically be terminated with a carriage return. Three special characters are used to accomplish this. They are:

```
& Will translate to FILENAME/EXT.PASSWORD:D **
| Will translate to FILENAME/EXT
; Will translate to <ENTER> (Note that the <ENTER> translation can be
  assigned to some key other than the ";" with the ENTER parameter).
```

\*\* The PASSWORD mentioned in this section will only be used in the version for LDOS 5.1. Those using LDOS/TRSDOS 6.0 can disregard any reference to it. The PASSWORD referred to will be EZTO, which is a substitute for the LDOS 5.1 master password.

For example, consider the following FM command:

```
FM data:0 :1 (FI,STR="RENAME & /OLD")
```

If the files DATA1/ARS and DATA2/ARS matched the command, the resulting output file would have the lines:

```
RENAME DATA1/ARS.EZTO:0 /OLD
RENAME DATA2/ARS.EZTO:0 /OLD
```

Once the FM command had completed, issuing a DO FMOUT/JCL would perform the desired renaming function.

#### E (Enter Character)

This parameter provides a means to designate a character that will be translated into a carriage return, or "Enter" character, in an output string. The default character value is a semicolon ";". If it is necessary to substitute some other character, the format E=value, or E="character" can be used. Specifying E=NO will cancel this feature.

## DRIVE NUMBERS

Since FM was designed to be a utility for file comparison, drive numbers play an important part in the final results of an FM command. For ease of explanation, this section will be broken up into four parts; the Display Mode, the Kill mode, the Move mode, and the Remove mode. In all modes, it is permissible to use more than one drive. Since the NEW and ABS parameters greatly affect the results of multiple drive commands, they will also be discussed in this section. The CK (Check before move) parameter will be detailed in the Move and Remove sections.

### IMPORTANT POINT

The FM utility normally operates with an implied "Old" parameter whenever two or more drive numbers are used in a command. This means that ONLY the files that exist on both drives will be considered as matching. The NEW parameter will override this feature in all four modes, and the ABS parameter may be used in the Move and Remove modes as explained in the following sections.

### IN THE DISPLAY MODE:

The normal use of a file display utility is to show the names of files on a particular drive. FM does support this standard feature. In addition, FM allows the display to be based on a comparison of one drive to another.

When using a single drive number, the FM display mode is similar to the normal DIR Library command. The files on the specified drive that meet any partspec criteria or parameters are displayed. The syntax of a display mode single drive command would be:

```
FM :d (parm,parm,...)
FM partspec:d
FM partspec:d (parm,parm,...)
```

As these general descriptions show, the display of files on a drive can be based on partial file specifications as well as other parameters. The important point to note: it is permissible to specify only one drive.

To use the comparison feature of FM, two drive numbers must be specified.

### IMPORTANT POINT

When using two drives on the command line, the resulting display will always consist of files from the FIRST DRIVE NUMBER specified.

The most basic use of two drive numbers in the display mode is to ascertain which files exist on both of the specified drives. For example:

```
FM :d1 :d2
```

As stated earlier, FM uses a default "Old" parameter. Thus, this command would display only the files on the first drive (d1) that were also on the second drive (d2). Partspecs and/or parameters could also be included in this type of command.

The NEW parameter will reverse this condition, allowing the display of files that exist on the first drive, but not on the second:

```
FM :d1 :d2 (NEW)
```

This command would show only those files that existed on the first drive, but were not found on the second. Again, partspecs and parameters could have been used.

As seen in both this example and the previous, the files displayed are always those that exist on the first drive number specified.

One of the most important uses of the Display mode is to preview the results of a Kill or Move command. Since those commands can greatly affect your files, the Display mode should be used to determine if the FM command to be used will produce the desired results. In the description of the Kill and Move parameters, use of the Display mode will be detailed.

#### **IN THE KILL MODE:**

The Kill mode of FM is used to remove files from a specified drive. Like the Display mode, either one or two drive numbers can be used. A single drive command will use only partspec information or parameters as criteria to determine which files will be killed. A two drive command will allow files on a second drive to be used for comparison purposes, but will still kill files on the first drive specified.

A single drive Kill command can have one of the following forms:

```
FM :d (KILL,param,param,...)
FM partspec:d (KILL)
FM partspec:d (KILL,param,param,...)
```

In these single drive examples, it should be obvious that the files to be killed are all on drive "d", and will be killed as long as they match the user entered partspec and parameters. When the Kill mode is used in this manner, it is very similar to the PURGE Library command, although it does provide added partspecing and parameter capabilities. The real power of the Kill mode can be seen when two drive numbers are used.

The syntax of two drive Kill command is:

```
FM :d1 :d2 (KILL,param,param,...)
FM partspec:d1 :d2 (KILL)
FM partspec:d1 :d2 (KILL,param,param,...)
```

In its basic form, this type of command means "Kill the files on drive :d1 only if they match my partspec and parameter criteria, AND ONLY IF THEY EXIST ON DRIVE d2." Thus, the FM Kill mode can be easily used to remove duplicate files from disks. As stated earlier, the NEW parameter can be used to reverse the comparison, allowing FM to kill files only if they DO NOT exist on the comparison drive. For example, the command:

```
FM partspec:d1 :d2 (KILL,NEW,param.... )
```

would remove FROM DRIVE D1, those files that matched the partspec and parameters but DID NOT EXIST ON DRIVE D2. The important point to note is that whether or not the NEW parameter is used, the files that match the criteria will always be killed on the first drive specified! The second, or comparison, drive will never be altered.

When using a two drive Kill command, there is a method to preview the results of a Kill without actually killing the files. By issuing the command without using the KILL parameter, the matching files will merely be displayed on the screen. Then if everything is correct, you can re-issue the command with the KILL parameter included. Note: when using the Display mode, the files are normally shown sorted alphabetically. In the Kill mode, the files are shown in the order in which they are encountered on the disk. If you wish the Display

mode to show the files in the order in which they will be killed, use the parameter (SORT=NO).

### **IN THE MOVE MODE:**

Move mode drive usage is slightly different from the Display and Kill modes in that it requires a minimum of two drive numbers, and will accept three. The NEW and ABS parameters will be discussed in this section, as they globally determine which files will be moved.

When using two drive numbers, Move is similar to the BACKUP utility using the (Old) parameter. The format of this type of Move command is:

```
FM :d1 :d2 (MOVE,parm,parm,...)
FM partspec:d1 :d2 (MOVE)
FM partspec:d1 :d2 (MOVE,parm,parm,...)
```

This type of command says "Move the files from drive :d1 to drive d2, based on any other criteria I have specified, IF AND ONLY IF THE FILES ALREADY EXIST ON DRIVE D2." This is unlike the normal Backup-by-files which moves files regardless of their existence on the destination disk. As an example, let us assume that drive 1 contains a disk with files you wish to move, and drive 2 contains a newly formatted disk containing no files. If you were to issue a "FM :1 :2 (MOVE)" command, no files would be moved to drive 2, since there are no existing files on drive 2! There are two ways to circumvent this default "old" parameter. One way is by use of the ABS parameter; the other is by using the NEW parameter (which invokes the opposite restriction).

#### The ABS parameter:

The ABS parameter is used to turn off the default "Old" parameter of the Move mode, and tells FM to move all matching files from the first drive to the second, regardless of their existence on the second drive. In this manner, FM will act like a normal backup by files command. The syntax of the command is:

```
FM :d1 :d2 (MOVE,ABS)
```

The use of a partspec and any additional parameters is optional. This command would tell FM to move all the files from drive "d1" to drive "d2", whether or not they already existed on drive "d2".

#### The NEW parameter:

In a two drive Move command, the NEW parameter means "Move those files from drive :d1 to drive :d2, IF AND ONLY IF THEY DO NOT ALREADY EXIST ON DRIVE :d2". The syntax of the command is:

```
FM :d1 :d2 (MOVE,NEW)
```

This type of command has many practical uses, such as updating an archive disk with new files. Any files that already exist on the destination disk will not be touched.

### **THREE DRIVE MOVE COMMANDS:**

The three drive Move command allows the movement of files from one disk to another, based on the results of a comparison with a third drive. One of the most practical uses of this type of command is to move groups of files from a large drive to smaller one, based on comparison to another smaller drive. In this manner, the file extension or date can be readily used to create very specific archive or backup sets of disks.

When using the Move mode of FM with three drive numbers, it is important to remember the following:

The first drive number is always the SOURCE of the files to be moved.

The second drive number is always the DESTINATION of the files to be moved.

The third drive number is always the COMPARISON drive to determine the files to be moved.

The basic three drive Move command syntax is:

```
FM :d1 :d2 :d3 (MOVE,param,...)
FM partspec:d1 :d2 :d3 (MOVE)
FM partspec:d1 :d2 :d3 (MOVE,param,...)
```

In the three drive Move mode of FM, the NEW parameter is valid although the ABS parameter is not. The three drive Move command will function as follows:

NEW parameter not used - only those files that exist on both drive "d1" and "d3" will be moved from drive "d1" to drive "d2".

NEW parameter was specified - only those files that are on drive :d1 but NOT on drive "d3" will be moved from drive "d1" to drive "d2".

As can be seen from these examples, the third, or comparison drive, will never be changed.

#### **IN THE REMOVE MODE:**

The Remove mode is a "Move then Kill" command. The Remove mode command line syntax is the same as that of the Move mode with the exception of using the REMOVE, rather than the MOVE, parameter. The drive usage is identical.

In two drive Remove commands, files are moved from the first drive to the second, and then killed on the first drive. Partspecs and parameters can be used to identify the files to be acted upon.

In three drive Remove commands, files will be moved from the first drive to the second based on a comparison of the third drive, and then the files that were moved will be killed on the first drive.

#### **The ABS parameter:**

As with the Move mode, the ABS parameter is valid only with two drive commands, and means "move all files that match the partspec and/or parameters from the first drive to the second, and kill every file on the first drive after it has been moved."

#### **The NEW parameter:**

Again, this parameter means the same to Remove as it does to Move. However, any matching files will be killed on the first drive after they have been moved to the second.

#### **FM PARTSPECING AND WILDCARDING**

Besides the ability to use multiple drives for comparison purposes, FM provides another very powerful file inclusion/exclusion feature by its use of partspecing and wildcard characters. A partspec, or "partial file specification", means all or part of a file specification. One of the main

uses of partspecing is to select a group of files that have similar names or extensions, or both. For FM purposes, a standard filespec is defined as:

FILENAME/EXT

FILENAME is from 1 to 8 alphanumeric characters, the first of which must be alphabetic. All files must have a filename.

/EXT is an optional extension of 1 to 3 alphanumeric characters, the first of which must be alphabetic.

(Alphanumeric - consisting of the letters A-Z, and the numerals 0-9. No other characters may be used.)

Since the four FM modes (Display, Kill, Move, and Remove) are all very similar in their command structure, the examples in this section will be valid no matter what mode is used. The next section of the documentation will explain the differences between using single versus separate filenames and extensions.

#### PARTSPECING: Use of Filenames

The FM utility allows the use of either a filename (inclusion) or NOT filename (exclusion) to be used to specify a particular group of files. The format for this type of partspecing is:

FILENAME (Inclusion - means include all files that match this filename)

-FILENAME (Exclusion - means exclude all files that match this filename)

A filename used as a partspec can be from 1 to 8 characters long. One very important point to remember is that checking for a match will be done only on the number of characters specified in your filename partspec. For example, if an FM command of CO:0 were used, all of the following files would match, assuming of course that they were on drive 0:

COMP/CMD CO CONFIG/SYS CONTINUE

As shown in this example, neither the number of characters AFTER the partspec nor the file extension are normally taken into consideration when matching a "filename only" partspec (this can be changed, however, as explained later in the Wildcard and Separate Extension sections). The main purpose of this type of filename partspecing is to choose a group of files that all start with a common prefix; a very handy feature when examining a disk that contains files from more than one application. The same method of matching holds true when using a filename as an exclusion spec. Thus, if an FM command of -CO were used, all files would be shown EXCEPT those that started with the letters CO.

It is also possible to specify only files that have no extensions. To do so, use a filename partspec as follows:

FILENAME/ or -FILENAME/

By using a "/" character with no following extension, FM assumes that only those files with no extensions should be matched. Again, the filename may be either an inclusion or an exclusion spec.

IN SHORT: Using a FILENAME or -FILENAME as a partspec will find a match with any file that starts with the partspec, and does not care what extension, if any, the file has. The FILENAME/ partspec is used to find files with no extensions.

NOTES: Much greater selectivity is possible by including wildcard characters in the filename partspec, or by including a separate extension, as will be explained later.

#### PARTSPECING: Use of Filename/Ext

Using a FILENAME/EXT partspec in an FM command is very similar to using a filename only. The only difference is that the file's extension must match the /EXT specified. It is important to understand that both a filename and extension will be evaluated the same way; only the number of characters used in the FM command will be considered when searching for a match. Thus a command such as CO/A would find a match:

```
CO/A   CO/ASM   CONFIG/ART   COMP/ASM
```

As this example shows, any characters in excess of those used in the command are disregarded, either in the filename or extension. Again, using the NOT character will exclude any files that match. Thus a command of -CO/A would show files except those that had a filename starting with CO, and that also had an extension starting with A.

SUMMATION: Using a FILENAME/EXT or -FILENAME/EXT as a partspec considers only as many characters as were specified in the filename and extension.

#### PARTSPECING: Use of /Ext

Just as files can be included or excluded according to their filenames, their extensions may also be used as the criterion. Again, only as many characters as are entered in the /ext partspec will be considered for matching purposes. Thus, using the FM command /C:0 would match files with extensions such as:

```
/CMD   /CCC   /COM   /CIM   /CE   /C
```

The more specific the partspec used in the FM command, the more specific the matching will be. Using the extensions displayed from the previous example, a partspec of /CC would only match the /CCC extension.

It is also allowable to enter a NOT extension partspec. For example, using an FM command of -/CMD would display all files EXCEPT those that had an extension of /CMD. A command of -/C would exclude any files that had an extension that started with /C, etc.

IN SHORT: The /EXT partspec is similar to the FILENAME partspec in that it uses only the number of characters entered when searching for a match in an extension, and that the NOT character can be used. However, the filename will not have any effect when using an extension-only partspec.

#### PARTSPECING: Using separate Filename and Extension

As the previous sections have demonstrated, anytime a single filename/ext partspec is specified, both the filename and the extension must match when searching for inclusion or exclusion files. The FM utility does, however, provide a way to specify a separate filename and extension in the same command. Moreover, either the filename or the extension or both may be NOT specifications. Refer to the following table, taking particular note of the comma separating the filename and extension:

```
FILENAME,/EXT  
-FILENAME,/EXT  
FILENAME,-/EXT  
-FILENAME,-/EXT
```

Using separate filenames and extensions is very similar to the normal partspecing previously described as far as searching for matches is concerned, using the normal FM "two part" procedure. The filename is first checked for a match, and then the extension. By using a comma to separate the filename and extension, it is possible to use either part SEPARATELY as an inclusion or exclusion criteria, totally independent of the other part. For example, suppose you had a disk in drive 1 that contained the following files:

```
FM10/ASM    FM11/ASM    FM1/CMD    FM1/TXT
FM/CMD      FM2/TXT      FM12/ASM
```

Looking at the file extensions, it appears that there are three types of files on this disk; /ASM assembler source files, /CMD object code files, and /TXT text documentation files. Also, the files all have filenames starting with the letters FM. To act on any one group of files, a command such as:

```
FM fm/txt:1
FM /txt:1
```

could be used to show all /TXT files. If it were desired to move only the /TXT files to another drive, a command such as:

```
FM fm/txt:1 :2 (MOVE)
FM /txt:1 :2 (MOVE)
```

could be used. Now consider the case where you want to move all of the files except the /TXT files. This could be done with the command:

```
FM -/txt:1 :2 (MOVE)
```

So far, the need for using the separate filename and extension feature has not been demonstrated. This is because of the small number of files that are the example disk. In normal practice, several groups of files are stored on the same disk. Consider the following revised display of a disk in drive 1:

```
FM10/ASM    FM11/ASM    FM1/CMD    FM1/TXT
CON/CMD      CON/TXT      TEST/ASM    TEST/TXT
```

In cases where multiple file groups are stored on the same disk, the separate filename and extension feature could be used to match specific groups of files:

```
FM fm,-/txt (Match all files starting with FM except those that have a /TXT extension)
```

```
FM -fm,/txt (Match all files with a /TXT extension except those that start with FM. In this case, CON/TXT and TEST/TXT would match)
```

```
FM -fm,-/txt (Match all files that do not start with FM and that do not have a /TXT extension. In this case, CON/CMD and TEST/ASM would match)
```

IN SHORT: The separate filename and extension are generally used when it is desired to match a group of files on a disk that contains many different file types.

## WILDCARDING:

Wildcard characters, or WCC, can be used in filename and extension partspecs to specify different groups of files. In FM, three WCC are allowed:

- \$ - The "mask" character
- \* - The "instring" character
- ! - The "truncate" character

The \$ WCC can be used in combination with the \* and the !. However, the \* and the ! are mutually exclusive.

### WILDCARD: The \$ character

The \$ mask character is used as its name implies; to mask, or disregard, character positions when searching for a match. Its function is identical to the normal LDOS \$ WCC. It is used primarily to search for a particular part of a filename or extension at a known offset. For example, consider the filespec:

TEST101/V21

This filespec would be matched by using any of the following partspecs in an FM command (These are just some of the possible partspecs):

```
$$ST1          $$$101          $$$101/V21 /$21
T$ST$01/$$1    TEST1/V$1
```

As can be seen from these examples, the \$ character is used as a mask, or "don't care" character. No matter what character is in the filename or extension at that point, it will always be considered a match. This type of wildcarding is very useful for files that have common groups of characters at a known place in the filename or extension. It should also be kept in mind if creating files for an application. For example, suppose a home checkbook balancing program used files with the filenames consisting of the first three letters of the month followed by the two digit year. A data disk for this program might contain the following files, among others:

```
JAN83/CKB    MAR83/CKB    APR83/CKB    FEB83/CKB
```

Using a partspec of \$\$\$83/CKB, or even \$\$\$83, would match these files. In cases where the desired field to match (in this case, the "83") is not at the same offset in the filename or extension, the \* WCC can be used to find a match.

Note that in none of the previous examples was the \$ ever used as the last character in the filename or extension. Although this would be permissible, it would serve no purpose whatsoever. Remember that normal partspecing only searches forward for as many characters as were entered on the command line (i.e. "FM1" would match FM101, FM15, FM1TEST, etc.), thus, any trailing \$ WCC would be unnecessary. If a specific LENGTH of filename or extension needs to be specified, see the ! WCC explanation.

IN SHORT: The \$ character is used to mask off, or cause a "don't care" condition for characters at certain offsets in the filename or extension. It should not be used as a trailing character.

NOTE: The \* and ! WCCs can also be used with the \$, and provide certain other matching features.

### WILDCARDING: The \* character

The \* is the "instring" WCC (instring means "in the string"). It is used to match a filename or extension partspec no matter where that partspec is embedded in the target filespec. For example, consider the FM commands:

```
FM *ba:1
FM /*ba:1
```

The first command means "match all files on drive I that have the letters "BA" anywhere in the filename, and would match files such as BASIC/CMD, TBA/CMD, MOMBA, NEWBAK/DAT. The second example says "match any file with the characters "BA" anywhere in the extension, and would match files like TEST/BAS, XMPL/TBA, ALLIN/BAK.

The length of the partspecs is limited to the \* plus 7 characters for a filename, and the \* plus 2 characters for an extension.

The Not character "-" can also be used with the \*, in commands such as:

```
FM -*ba:1
FM -/*ba:1
```

These types of commands would exclude any files that had the characters "BA" anywhere in the filename or extension. As listed in the PARTSPECING section of the documentation, the following FM commands can also use the \* character:

```
FM -*filename/ext
FM -*filename,/*ext
FM *filename,-/*ext
FM -*filename,-/*ext
```

The \* WCC can also be used with the \$ WCC. This type of implementation is best described through example:

```
FM *$$te
FM /*$a
```

In the first example, the command states "match those files that have a 4 character string ending in TE anywhere in the filename". Files such as SEPTEMBER/82, MYTEST, etc. would match. The second example means "match any file with a two character string ending with A in the extension." Files such as TEST/BAS, XMPL/TBA, etc. would match. Practical uses of the \* and \$ combination will depend upon the files that are to be manipulated. When creating files (such as with a word processor, mailing list program, etc.), keep in mind this feature of FM and make file groups that are easily displayed, moved, or killed.

IN SHORT: The \* character is used to provide an instring, or sliding field type of comparison function for matching filenames or extensions. It can be used along with the \$ WCC.

### WILDCARDING: The ! character

The ! character is used to truncate, or cut off, the search for a match of a filename or extension partspec. Its primary use is to specify a particular length of partspec. It is best demonstrated through example; consider the following FM command using a truncated filename:

```
FM te!
```

This command would match files such as:

TE/CMD        TE            TE/BAK        TE/X3

but not the files:

TEST            TERRA/101    T/BAK

As the example shows, this type of partspec is used to match files that have an EXACT length of two characters, with those two characters being "TE". Any files that with filenames shorter or longer than two characters will not match. The same will be true if the partspec is an extension:

FM /CM!

This command would match files such as:

TEST/CM        MYCMD/CM     NOWAY/CM     A/CM

but not files like:

TE/CMD        OVLY/C        DATA/CMA    READ2/COM

As shown, only those files that have exactly two characters in the extension, and those two characters being "CM", will match.

The Not character "-" may also be used with the WCC to specify an exclusion partspec. Commands such as:

FM -te!  
FM -/te!  
FM -this,/c!

are all acceptable, as well as the other command forms described in the previous Partspecing sections.

To specify all files that have a definite length of filename or extension, the \$ WCC can be used along with the ! WCC in the following manner:

FM \$\$! (All files with exactly two characters in the filename, regardless of any extension)

FM \$\$!/ (All files with exactly two characters in the filename, as long as they have no extension)

FM /\$\$! (All files with exactly two characters in the extension)

The Not character "-" can also be used to exclude matches when using the ! WCC, such as:

FM -TE! (Exclude any files that have TE as the only characters in the filename)

FM        (Exclude any files that have a single character extension)

FM        (Exclude any files that have a single character filename)

IN SHORT: The ! character is used to specify an exact length of filename or extension when searching for a match. It can be used along with the \$ character.

## FM PARAMETER DESCRIPTIONS:

Besides the mode parameters discussed earlier, FM has many other parameters to help select a certain group of files, perform directory updating functions, or to let FM run under control of the LDOS JCL (Job Control Language).

### DISPLAY FORMAT PARAMETERS:

#### A (Allocation):

The A parameter is used with the Display mode (the other modes automatically turn this parameter on). The normal Display mode will show the files in alphabetical order listed across the screen, such as:

```
FILENAME/EXT      FILENAME/EXT      FILENAME/EXT      FILENAME/EXT
```

If the A parameter is specified, the files will be shown one per line, with other information following the filename and extension. The format would be:

```
FILENAME/EXT |MM/DD/YY| 0000 R  
-----|-----|-----  
aaaaaaaaaaaa *bbbbbbbb* cccc d
```

aaa This field will be the filename and extension.

bbb This field is the file's Mod date. This is the date of creation, or the last date the file was written to.

\* The asterisks represent the "mod status" characters. If the file has been modified (written to) since it was last backed up, then the characters in these two positions will be plus signs "+". Otherwise, the vertical bars "|" will be used as separators.

ccc This field is the total number of "units" in the file. If not changed by specifying the G or K parameter, this will be the total number of disk sectors assigned to the file.

d This field will show whether the normal "R" (number of records) mode was used, or if the G or K parameter was used. It will appear as the letter R, G, or K.

Normally, a Display mode command is done without using the A parameter. However, if it is necessary to see the additional information, the A parameter can be specified. If the P (send output to printer) parameter is specified along with the A parm, the printer output will also be in single line format, and will contain the date and unit information.

#### G (Gran) or G=:

The normal "unit" display of FM is in Records, meaning the number of disk sectors used by a file. The G, or Gran, parameter is used to specify the output as the number of grans assigned to a file. The main use of this parameter will be when dealing with disks of different sizes (311, 511, 8", hard) or different densities (single, double, hard disk format). Since a gran represents the minimum number of sectors that can be allocated to a file, consider the following table which shows the minimum number of sectors (DDEN=Double density, SDEN=Single density):

5" Floppy, SDEN	- Gran = 5 sectors, 1.2K
5" Floppy, DDEN	- Gran = 6 sectors, 1.5k
5" Hard,	- Gran = 16/32 sectors, 4/8K
8" Floppy, SDEN	- Gran = 8 sectors, 2K
8" Floppy, DDEN	- Gran = 10 sectors, 2.5K
8" Hard,	- Gran = 16/32 sectors, 4/8K

As seen in this table, there is a wide variance in the number of sectors, or records, that make up a granule. Thus, a file that contained only 1 record would take up 1 gran on a disk; however, the total disk space used would vary greatly.

The G parameter should be entered followed by a value representing the number of sectors per gran. Referring to the previous table, it can be seen that the value for a 5", SDEN Floppy would be 5, and for an 8" SDEN Floppy would be 8, etc. IF the G parm is used with no value, 6 will be assumed (matching a 5" DDEN Floppy).

When files are to be moved from one type of disk to another, the G parameter can be used to show the total number of grans that will be needed. Whenever the Display mode is used, the ending line always will show the total number of units taken up by the matching files. If the G parameter is used, this unit total will be the number of grans needed to store the files on a disk of the type having that gran size. By examining the final display of total grans needed, and comparing it with the number of free grans available on the disk, it can easily be determined if the desired files can be moved. The LDOS Library command FREE, when used to show the free space map of a single drive, will also show the free granules available on that drive.

#### K (1024 byte blocks):

Like the G parameter, K can be used to shift the FM output display from the normal Record mode to show the number of K, or 1024 byte blocks a file uses. The number of K will be based on the actual number of records in a file, not on the space allocated to it on the disk. No partial K will be shown, and any remainder will be rounded up to the next higher K. For example, a file with six records takes up 1.5K, but will be shown as 2K.

The main use of the K parameter is to help determine the actual space used by files in a system with many different drive types. Like the G parameter, this can help determine the actual amount of space needed when moving files between drives. The total number of K for all matching files will be shown in the final display line. If the A (Allocation) parameter is used, the K for each individual file will also be displayed.

#### O (Sort):

The O parameter is used to turn off the normal alphabetic file Display mode sort. To use it, issue a command such as:

```
FM :0 (O=OFF)
```

Although the Display mode normally shows files sorted alphabetically, the other modes act on files as they are found while scanning the directory. To allow the files to be displayed in the same manner, the O parameter can be turned off. This may be desirable to preview the results of a Kill, Move, or Remove command, seeing the files in their actual order.

#### NOTE

Sorting the file specifications requires a certain amount of free memory. If there is not enough memory available, FM will not display the files in alphabetical order. This will be a very unusual occurrence, but can happen if free memory drops below a certain point.

## P (Printer)

The P parameter provides a way to send the results of an FM command to a printer. In the Display mode, the files are normally sent to the printer four across. Each entry sent will include the filename, the extension (if applicable), and the drive number of the source drive. If the A parameter is also used, each file will be printed on a separate line. The file's date, modification status, and size will also be printed. The size will normally be in total records, but can be changed with the G or K parameters. If the P parameter is used in the Kill, Move, or Remove modes, the file display will also be sent to the printer as if the A parameter was also used.

## Q=ON/OFF (Query)

The Q parameter affects the four FM modes differently. In the Display mode, Query is normally on. A Q=OFF parameter will keep the screen display from pausing as it normally does every 11 lines, and will display all files non-stop. If the JCL parameter was used on the command line, it will automatically set Q=OFF for Display mode commands.

With the Kill, Move and Remove modes, the Q parameter lets the user be queried whether to act on the currently displayed file. For Kill and Remove, the default is ON (query before a file is killed); for Move the default is off (move files without asking). The default may be overridden by specifying the desired state of the Q parameter on the command line. In the Kill, Move, and Remove modes, the following prompt will appear if Query is ON:

(Y/N/C) ?

Pressing <Y> tells FM to act on the file; pressing either <N> or <ENTER> means to bypass the file without action. pressing <C> will act on the file and turn OFF the Query function. FM will continue nonstop from this point.

The user is cautioned against using the Kill or Remove modes with the Q=OFF parameter until familiar with the way FM operates. Killing files automatically is an easy way to clean up a disk, but can be disastrous if the wrong files are involved.

## **FILE ATTRIBUTE PARAMETERS:**

The next group of parameters lets the user specify groups of files based on their directory attribute type or modification status, as well as by a user specified flag bit. Certain of these parameters use existing information about a file, while others allow a file's information be changed.

Two parameters, SM (Set Mod flag) and CM (Clear Mod flag) are provided to either set or clear the modification flags on specified files. Three parameters deal with the "Flag bit". This bit is also used by the PDS utility (a product of MISOSYS) to indicate PDS files. LDOS indicates files with this bit set by displaying an asterisk after their name in a DIRectory display. FB will find all files with the Flag bit set, SFB will set the Flag bit on matching files, and CFB will clear the Flag bit. By setting either the mod flag or the user flag bit, files ordinarily dissimilar in filename, "extension, or other attributes may now be easily manipulated as a group.

The PR parameter will find all files matching a specified protection level, 0-7 (corresponding to the levels as described in the LDOS library command ATTRIB). Files may also be selected based on their visibility and system status.

### INV (I)

The INV, or I, parameter means act on only those files that are marked in the directory as being INVisible. These normally would be LDOS Utility files. To view the invisible files on a disk, issue an FM :d (I) command. One of the reasons for a user to make a file invisible is to prevent directory display from appearing cluttered. While using this parameter will find invisible files, it will not change them to visible files. That function can be performed with the regular LDOS Library command ATTRIB.

The I parameter can be used in conjunction with the VIS and SYS parameters as required to search for all three file types.

### VIS (V)

The VIS parameter means act on only those files that are visible in a directory display. If neither INVisible nor SYStem are used as parameters, FM will use VIS as a default. In normal practice, the only files that are not marked as visible will be the operating system files and utilities, or special user files. Since most file maintenance is done on visible user programs and data files, and VIS defaults to on, this parameter will normally not have to be specified.

If it is necessary to deal with all file types, such as when making a copy of a system disk, use a command such as (V,I) meaning both visible and invisible, (V,S) meaning visible and system files, or (V,I,S) meaning all file types.

### SYS (S)

The SYS parameter is used to specify LDOS system files. It can also be used along with the VIS and INV parameters to specify file groups as needed. Its main use is to allow the creation or updating of system disks.

When using the S parameter in the Move or Remove mode, FM will attempt to correctly place LDOS system files in their required directory slots. If that directory slot is in use by some other file, FM will abort with an error message. This should not happen with the 5.1 version operating system, but may occur with the 6.0 version. To avoid any problems, the S parameter should only be used to move system files to either freshly formatted disks, or to existing system disks.

If the SYS0/SYS file is moved, FM will also move the System Information Sectors as described in the operating system's technical documentation. This means that the state of any sysgened configuration, the state of the power-up date and time prompts, and the default drive configurations will be moved to the destination disk.

### CM (Clear Mod flag)

In the Display mode, the function of the CM parameter is very straight forward. It removes the modification flag from all matching files on the source drive.

The CM parameter cannot be used in the Kill mode. It would make no sense to clear a file's mod flag when it is about to be removed from the disk.

In the Move mode, using CM will always clear the mod flag on the source and destination drives. For this reason, the source drive cannot be write-protected, or the FM command will abort with an error.

In the Remove mode, using CM will clear the mod flag on the destination drive. As with Kill, nothing will be done to the source drive, because the matching files there will be removed from the disk.

### SM (Set Mod flag)

In the Display mode, using the SM parameter will set the mod flag on every file that matches the specified partspec and/or parameters on the source drive. One of the main reasons for setting mod flags is to be able to then use the MOD parameter to move these files from disk to disk. For example, suppose a disk contained several files to be moved, but the files did not share common characteristics such as a unique extension, filename, or mod date. You could use FM to set the mod flag on these files, and then use an FM :d :d (MOVE,MOD) command to transfer the files to another disk. This procedure could be repeated as many times as necessary, because the mod flags will not normally be cleared by the Move mode.

The SM parameter is not valid in the Kill mode, because any matching files will be removed from the disk.

In the Move mode, using SM will set the mod flag on both the source and the destination drive. For this reason, the source drive cannot be write-protected, or the FM command will abort with an error.

In the Remove mode, the destination mod flag will be set. Since the source file will be killed, the SM parameter does not consider it.

### FB (Use Flag Bit)

The "Flag" bit is a special mark in a file's directory entry. It can be set or cleared with the SFB and CFB parameters, respectively. Files that have their Flag bit set will appear in a normal operating system DIR Library display followed by an asterisk. The main use of the Flag bit is to allow a group of files to be marked, whether or not they have any other common filespec or attributes. This will facilitate moving or killing these files as a group, without the need to issue many separate commands.

### CFB (Clear Flag Bit)

This parameter provides a means to clear the user "Flag" bit discussed in the FB and SFB parameters. To see files that have this flag bit set, do a DIR Library command and observe those files that have an asterisk after the filespec.

In the Display mode, using the CFB parameter will clear the Flag bit on any matching file on the source drive. It, of course, is not valid in the Kill mode, as any matching files will end up removed from the disk.

In the Move mode, the Flag bit will be removed from both the source and destination disks. Be sure that the source disk is not write-protected, or the command will abort.

In the Remove mode, the Flag bit will be cleared on the destination disk.

### SFB (Set Flag Bit)

This parameter provides a means to set the user "Flag" bit for a file. It can be used as a means of marking a group of files to be manipulated in a common manner. To see files that have this flag bit set, do a DIR command and observe those files that have an asterisk after the filespec.

The section on the SM (Set Mod Flag) parameter gives a discussion of why you may want to use a special method of marking files.

## PR (Use PROTection Level)

This parameter provides a way to specify files by their assigned operating system protection level. These levels are different between the 5.1 and 6.0 versions, so check the DOS manual to determine what level names correspond to which protection level numbers. A file's protection level can be changed with the ATTRIB Library command.

In the Display mode, this parameter limits the display to those files that have a matching protection level. Normally, this protection is linked with the use of a password assigned to a file. The different modes will act on only those files that match the protection level.

The PR parameter is used with a value 0 to 7, with 0 being the least protection (FULL access, usually the level on user programs and data files), and 7 meaning the greatest protection (NO access, normally used only on operating system files). If, for example, a disk contained program files protected as Execute Only, these files could be accessed with the command:

```
FM :d (PR=6)
```

Thus by using the ATTRIB Library command to set common protection levels on files, they may be manipulated as a group by FM.

## **DATE SPECIFICATION PARAMETERS:**

Three FM parameters deal with a file's date; two of them (DATE and TODAY) are used to select files based on a date, while the third, (SD), can be used to change a file's date. The term "date" when used in reference to a file is the mod date - the date the file was created or last written to. To view the current mod date for any file, the FM Display mode or the normal DIR Library command can be used, providing the A parameter is used in either case. For example:

```
FM TEST:0 (a)
DIR TEST:0 (a)
```

Either of these two commands will show all files starting with TEST, and will include the file's mod date, as well as other information. For a further explanation, see the FM section on the A (Allocation) parameter.

## DATE (D)

The date parameter is used to specify a single date or a range of dates that will be used to determine inclusion of a file. Four forms of the parameter are allowed:

D="MM/DD/YY" Files only with the specified date.

D="M1/D1/Y1-M2/D2/Y2" Files with dates between the first and the second, inclusive.

D="-MM/DD/YY" Files with dates less than or equal to the specified date.

D="MM/DD/YY-" Files with dates greater than or equal to the specified date.

The D parameter may be used in all modes. For example, assume that the current date is 07/10/83. To find all files that have been written to in the last four days, one of the following commands could be used:

```
FM :d (D="07/07/83-07/10/83")
FM :d (d="07/07/83-")
```

This command would result in a display of all visible files that had mod dates of July 7th through July 10th, the current date. This same date parameter could now be used move or kill selected files in the group.

SD (Set Mod date)

The SD parameter is use to set the modification date of specified files. The date can be set either to the current system date or to a user specified date. The format for this parameter is:

```
FM :d (SD) This command will use the current system date.
```

```
FM :d (SD="MM/DD/YY") This command will use the date string specified by
MM/DD/YY. The restriction on the range of acceptable dates are the same as
those for the operating system.
```

This parameter is not valid in the Kill mode.

In the Move mode, using SD will set the date on the destination drive files. The files on the source drive will retain their original dates.

In the Remove mode, the date will be set on the destination drive files. Nothing will be done to the source drive, as the matching files there will be killed.

TODAY (T)

One of the most common uses of the date parameter is to deal with files that have been recently modified. In recognition of this, the TODAY, or T, parameter has the ability to define a range of dates relative to the current setting of the LDOS system date. It is merely a shorthand form of the DATE parameter. The allowable range is from seven days previous, to the current date. The format of the T command is:

```
T This command means files with today's date.
```

```
T="n" where "n" is a number 1 to 7; 1 meaning yesterday and 7 meaning seven
days ago.
```

```
T="n1-n2" where n1 is greater than n2. This command means "from n1 days ago
to n2 days ago".
```

```
T="-n" This command means "from seven days ago to n days ago".
```

```
T="n-" This command means "from n days ago to today".
```

The T parameter can be used in all modes. Again, remember that it defines files with dates matching the specified range, and is merely a convenient shortening of the DATE parameter.

#### **FILE MODIFICATION PARAMATERS:**

The file modification parameters are used to select either files that have been modified or files that have not been. Generally speaking, the mod flag indicates that the file was written to since it was last backed up with the operating system BACKUP Utility. To determine if a file is considered modified, use an FM Display mode command with the A parameter, or a DIR Library command. In either case, a Plus sign "+" will appear after every modified file. The FM parameters CM and SM also can be used to clear or set a file's mod flag.

### MOD (M)

The MOD parameter can be used to specify only those files that have their "Mod" flags set. In normal LDOS operation, a file's mod flag will be set any time a file is written to or changed in any manner. As explained previously, the SM parameter of FM can also be used to set the mod flag on selected files. Some of the reasons for using mod flags other than the way the operating system normally does are mentioned in the SM parameter section.

### U (Unmodified)

This parameter can be used to specify only those files that have not been written to or changed. It, in effect, performs the opposite function of the MOD parameter. As with the MOD parameter, the SM and CM parameters can be used to change a file's Mod status.

### **FILE SIZE PARAMETERS:**

Two parameters, Limit and Size, can be used very effectively in systems where it is necessary to move data from large volume drives to smaller ones. While the FM utility does not provide a way to move a single large file onto a series of smaller disks, it does give the user an easy means to bypass these large files in Move or Remove commands, and then see which files were not moved.

### LIMIT (LM)

The Limit, or LM, parameter is used to specify a maximum file size. The value for the Limit parameter will usually reflect the size of the destination disk to be used in the -Move mode, and must be expressed in K (1024 byte blocks). For example, a 40 track, double density, single sided disk has 174K free after formatting. A 35 track, single density, single sided disk has approximately 83K free after formatting. Attempting to move a single file that was larger than the free space on the destination disk would cause an endless series of "Disk Full - Swap Disks" requests, and the Move would not proceed past that point. To circumvent this problem, you can use the Limit parameter to instruct the FM utility to ignore any files that will not fit on a single destination disk. For example:

```
FM :0 :1 (MOVE,LM=174)
```

This command tells FM not to attempt to move files larger than 174K, the maximum available space on a 40 track double density floppy. Once the Move has finished, the Size parameter can be used to display any files that were above the limit.

### SIZE (Z)

The Size, or Z, parameter is used to view files that are larger than a specified length. This parameter can be used in conjunction with the Limit parameter when moving files from a large volume drive to multiple smaller disks. The format for the parameter is:

```
SIZE=nnn
```

The "nnn" is the size in K (1024 byte blocks). For example, using a SIZE=174 will display the files that contained more than 696 record blocks. Like the limit parameter, SIZE is primarily useful when moving files from large volume disks to smaller ones.

## JCL PARAMETERS:

The JCL (Job Control Language) parameters provide a means to run the FM utility under the control of the LDOS JCL processor. The use of these parameters is necessary due to the automatic error recovery feature of FM. During certain operations, FM will prompt the operator if a disk I/O error has occurred, asking whether to retry the operation, skip the file in question, or abort the operation entirely. This unexpected prompt could cause an active JCL to get "out of sync", and random results could follow. To avoid any problems, the following two parameters can be used.

### JCL=retries

This parameter tells FM that the current operation is being controlled by a JCL file, and/or lets the operator specify the number of automatic retries to be attempted in case of an error. Normally, a specific value is used, and can range from 1 to 10. The format of a command would be:

```
FM :0 :1 (MV,JCL=3)
```

If JCL is used with no value, two retries will automatically be done. In either case, if the retries are not successful, the file will be skipped and the FM operation will continue. The one exception to this is the occurrence of a "Disk full" error, which will abort an FM command running under JCL control.

Even when not running under JCL control, the JCL parameter can be used to specify a number of automatic retries, in case a the operator will be away from the computer during the operation, and does not wish to be prompted on errors. This will assure that the operation will complete. The FM final display message will include an extra line if any files were skipped when running with the JCL parameter active.

### ABORT

The ABORT parameter can be used to override the automatic retry function that occurs when running under JCL control. If ABORT is specified, the occurrence of any error will abort the FM command, and the JCL processing.

This parameter should be used when it is desirable to abort an operation no matter what type of error is encountered, such as when making copies of very critical data.

## OUTPUT FILE PARAMETERS:

The output file parameters of FM are ENTER, FILE and STRING. They provide a means of writing any matching filespecs to an ASCII disk file. If desired, the filespecs can be embedded in a larger ASCII string, with the entire string being written to an output file. The main purpose of this feature is to create JCL files that can later be executed to perform a desired function on a group of files.

### FILE="filename"

The FILE, or FI, parameter provides a way to generate an optional ASCII file containing certain user selected information including filename and/or drive numbers during an FM command. The entire filename and extension can be entered on the command line, or the FILE parameter can be used by itself. The results will be as follow:

```
FI="FILENAME/EXT" would use the user specified filename and extension.
```

```
FI="FILENAME" would use the user's filename and add the /JCL default extension.
```

```
FI with no value will default to a file named FMOUT/JCL.
```

If an error condition is encountered when writing the output file, a single error message will be displayed but the FM command in progress will continue. If this happens, the output file should be considered fatally flawed, and should NOT be used until it is examined. Probably the best thing to do is to kill it, and try the FM command again.

STRING="desired output string"

The STRING parameter can be used to specify the line sent to an output file. It allows for insertion of matching filespecs with and without drive number and password, and can embed <ENTER> characters in the string. Each string sent will automatically be terminated with a carriage return. Three special characters are used to accomplish this. They are:

& Will translate to FILENAME/EXT.PASSWORD:D \*\*

| Will translate to FILENAME/EXT

; Will translate to <ENTER> (Note that the <ENTER> translation can be assigned to some key other than the ";" with the ENTER parameter).

\*\* The PASSWORD mentioned in this section will only be used in the version for LDOS 5.1. Those using LDOS/TRSDOS 6.0 can disregard any reference to it. The PASSWORD referred to will be EZTO, which is a substitute for the LDOS 5.1 master password.

The STR parameter used without any value (i.e., a command such as FM :0 (FI,STR)), will force the output string to consist of only the filespec, password, and drive number of the source drive. Otherwise, any characters included between the quotes will be sent to the output file, with the special characters being translated as described. For example, consider the following FM command:

```
FM data:0 :1 (FI,STR="RENAME & /OLD")
```

If the file DATA1/ARS matched the command, the resulting output file would have the line:

```
RENAME DATA1/ARS.EZTO:0 /OLD
```

Once the FM command had completed, issuing a DO FMOUT/JCL would perform the desired renaming function.

E (Enter Character)

The E parameter provides a means to designate a character that will be translated into a carriage return, or "Enter" character, in an output string. This assumes, of course, that the STR parameter was also used. The default character value is a semicolon ";". If it is necessary to substitute some other character, the format E=value, or E="character" can be used. Specifying E=NO will cancel this feature.

The main use of the semicolon or other ENTER character is to create a multi-line output string for later use. For example, if an FM command was used to find several files which were to be copied to another disk and then renamed, the following output string could be used:

```
FM TEST/NEW:0 (fi,str="copy & to |:1;rename |:1 /old")
```

Assuming that two matching files, TEST1/NEW and TEST2/NEW were found on drive 0, the resulting lines written to the output file would be:

```
COPY TEST1/NEW.EZTO:0 TO :1
RENAME TEST1/NEW:1 /OLD
COPY TEST2/NEW.EZTO:0 :1
RENAME TEST2/NEW:1 /OLD
```

Any number of embedded ENTER characters can be used, up to the point where the FM command will no longer fit on a single line.

#### **FINAL MESSAGE DISPLAY:**

When an FM command is finished, an appropriate message will be displayed. The exact message will depend on the mode (Display, Kill, Move, or Remove) that was selected, as well as the type of display requested (show Grans, K, or number of Records). In any case, the last segment of the sign-off message will be the total requested allocation of all files that matched the FM command line specifications and parameters. When it is desired to move files from one drive type to another, the total count from a display mode command can be used to determine how many destination disks will be required.

FM processing can be aborted at any time by pressing the <BREAK> key. If this is done, you will see the message:

```
Manual abort requested!
```

This lets you know that FM aborted before the end of its processing at your request. If FM finishes the request procedure, one of the following messages will be displayed, depending upon the mode selected.

```
Completed with AAA of BBB being matching files, CCC Total
Completed with AAA of BBB being new files, CCC Total
```

```
Completed with AAA of BBB specified files moved, CCC Total
```

```
Completed with AAA of BBB specified files removed, CCC Total
```

In these messages, the "AAA" represents the number of matching files that were found or selected. The "BBB" represents the total files on the disk. (Note: the LDOS files BOOT/SYS and DIR/SYS are never considered in any FM command, and are NOT included in the "BBB" count). The "CCC Total" message will indicate the total Records, Grans, or K of the matching files. The first two messages will come at the end of a Display mode function. The others will appear after a Move, Remove, or Kill command.

If the JCL parameter was used, and I/O errors caused a file to be skipped, the following message will be displayed after the total line:

```
CAUTION - files bypassed in JCL mode!
```

#### **ERROR CONDITIONS AND RETRY CAPABILITY:**

When FM is performing a file display, a disk I/O error will abort the command. If this happens, the appropriate LDOS error message will be displayed, with an exit to the LDOS Ready prompt following. No final display message will be shown.

If an error occurs in the Move or Remove modes, FM will not automatically abort, but instead will give the operator a chance to retry the operation, skip over it, or abort as desired. This will be done regardless of a Query On/Off condition. Included in the error prompt message will be the drive the error occurred on, the operation FM was trying to perform at the time, the LDOS error number, and what options the user has.

Open error, Source \*\* nn, <R>etry, <S>kip <BREAK> Abort  
Open error, Dest. \*\* nn, <R>etry, <S>kip, <BREAK> Abort  
Read error, Source \*\* nn, <R>etry, <S>kip, <BREAK> Abort  
Write error, Dest. \*\* nn, <R>etry, <S>kip, <BREAK> Abort

nn = The LDOS error number

If one of these error messages appears, there are three options available. Pressing <R> will retry the operation. The <S> option will skip over the current file and continue on with the operation. <BREAK> will abort the FM command entirely. To avoid aborting an FM command running under JCL control, the JCL parameter will assign a certain number of automatic retries. Once that number has been attempted, an automatic skip of the file will occur, and the operation will progress.

#### DISK FULL

One exception to the error trapping is a "Disk full" error during a Move or Remove command. If a disk full error is encountered, the following message prompt will be displayed:

Destination disk full replace it and press <ENTER>

If this message appears, remove the destination disk, replace it with the desired disk, and press <ENTER>. To abort the move, press <BREAK>. In either case, the last file displayed on the screen (the one that was caused the disk full error) will have been killed off the destination disk. If FM is being controlled by a JCL procedure, the occurrence of a disk full error will abort the JCL processing and display the appropriate error message.

## FM HELP SCREEN DISPLAY:

FM has a built in HELP! display function. If incorrect partspecs, drive numbers, or parameters are used, the offending command line will be shown, along with the Brief Help Screen listing. This Brief Help listing shows the available parameter abbreviations and wildcard characters only:

FM - Conditional File Exam/Kill/Move Utility - Ver. x.x.x

Command:

WILDCARD CHARACTERS \$=Mask, \*=Instring, !=Truncate

A, CK, P, Q, O, G, K, N, M, U, V, I, S  
D, T, SM, CM, SD, Z, LM, PR, SFB, CFB, FB  
JCL, ABORT, ABS, MV, KI, RMV  
FI, STR (&, |), E

The main function of this short Help display is to show the parameter abbreviations to help jog the memory. Note: the command FM (HELP,P) will send the full Help display to the printer rather than to the video. The display is:

FM ps:D (PARMS) or FM ps:D :D (PARMS) for Display Mode  
FM ps:D (PARMS) or FM ps:D :D (PARMS) for Kill Mode  
FM ps:D :D (PARMS) or FM ps:D :D :D (PARMS) for Move/Remove

WILDCARD CHARACTERS: \$=Mask, \*=Instring, !=Truncate

DISPLAY: A=Allocation, P=Printer, O=Sort, G=Grans, K=1024 bytes  
I=Inv, S=SYS, V=Vis, M=Mod, U=Unmod, FB=Flag bit, PR=Prot level

SM=Set Mod, CM=Clear Mod, SD=Set Date  
SFB=Set Flag bit, CFB=Clear Flag Bit

DATE: D="mm/dd/yy-mm/dd/yy", T=Today, or T="Then-Now"

Q=Query before action, N=New files only, ABS=Disregard Old  
CK=Check Dest. space when Moving w/New

SIZE: LM=Limit to spec. K, Z=Show over spec. K

AUTO: JCL=Auto retries, ABORT=Quit on error

FILE OUTPUT: FI="Output File Name", STR="Output string"

STR sub. chars: &=FILENAME/EXT.PSWD:D, |=FILENAME/EXT  
E=Character Value for embedded Carriage Return