

HDE, INCORPORATED

```
* * * * *  
*  
*      USER'S MANUAL      *  
*      -----            *  
*      HDE FILE ORIENTED  *  
*      DISK SYSTEM - FODS  *  
*      -----            *  
*      FOR THE SYM COMPUTER *  
*  
* * * * *
```

AUGUST 1980

COPYRIGHT AUGUST 1980

HUDSON DIGITAL ELECTRONICS, INC.
P.O. BOX 120
ALLAMUCHY, N.J. 07820

PRICE: \$7.50

REVISION RECORD

REVISION	DATE	PAGES	COMMENTS
REV 0	AUGUST 1980	ALL	INITIAL ISSUE

CONTENTS

INTRODUCTION TO FODS

LIMITED WARRANTY

FODS REVISIONS AND CHANGES

CARE OF DISKETTES

SYSTEM REQUIREMENTS

LOADING AND INITIALIZATION

DISK INDEX

FILE CONCEPT

DISK OPERATIONS - GENERAL

DISK OPERATIONS - SINGLE DRIVE SYSTEMS

DISK OPERATIONS - DUAL DRIVE SYSTEMS

BOOTSTRAP ROUTINE

SYSTEM COPIES

FODS ENTRY

ZERO PAGE USAGE

RAE OPERATIONS

ERROR NOTIFICATIONS

COMMAND STRUCTURE

APPENDIX 1 - FODS BOOTSTRAP

INTRODUCTION TO FODS

The HDE File Oriented Disk System is a random access, sequential file system designed for general applications. FODS has been developed to provide to the user a capability for performing file level disk transactions with ease. Files are transferred between disk and memory through simple, but powerful, commands which in most cases merely specify the action desired and the file name. User unique functions and routines are readily integrated into the command structure allowing system expansion limited only by available disk storage.

The HDE File Oriented Disk System (FODS) features:

- * Random file access
- * Sequential file storage
- * A powerful command structure
- * User routine integration into the command structure
- * Indexed (all) and non-indexed (OMNIDISK 65/8) storage

LIMITED WARRANTY

The HDE File Oriented Disk System has been carefully designed to assure correct operation when applied as described within the limitations identified in this manual. The nature of computer program development prohibits any claim that the FODS is error free. Therefore, this program is distributed with a "Limited Warranty".

- * Any errors discovered will be corrected in the first 90 days.
- * Catastrophic errors will be corrected by automatic mailing to all registered owners.
- * Minor errors will be corrected by request.

In any case, this warranty is limited to replacement of the disk and/or manual. No liability for consequential damages can be assumed or is implied.

If you believe you have discovered an error, document the conditions concerning the error, what you think is wrong, the version number and serial number of your copy and mail to:

HDE, inc.
P.O. Box 120
Allamuchy, N.J. 07820

Phone: (201) 362-6574

FODS REVISIONS AND CHANGES

HDE may, from time to time, revise all or part of the system computer programs to improve operation of existing functions and to add or delete capabilities as determined by the needs of our customers. HDE attempts to maintain a high level of standardization within the FODS user community. In order to approach this goal HDE, as a matter of policy and standard practice, announces release of such revisions by mail to all FODS users that have returned the PRODUCT REGISTRATION card included with each system. Each announcement includes a description of the modifications included within the change and the terms and conditions for update. It is the user's responsibility to request an update within the terms and conditions specified.

CARE OF DISKETTES

Although forgiving of considerable mistreatment, flexible diskettes must be handled with care. Several precautions should be strictly observed to preserve reliable operation.

- * Use quality diskettes. The FODS has been provided to you on a quality product that will provide reliable service for a considerable period of time. Nothing can be more distressing than to lose hours of labor because of a diskette failure. "Bargain diskettes" may prove to be extremely expensive in the long run.
- * The use of contact labels on the diskette jacket is acceptable and these are occasionally supplied with diskettes by some distributors. Once a label has been applied to a diskette, do not write on the label with anything but a soft, felt tip pen. Pencils and ball points must not be used.
- * Avoid any contact with the exposed disk surfaces. The diskette always should be in its protective sleeve any time it is not in the drive.
- * Store disks vertically, in a protective container. Do not place anything on top of a diskette, except another diskette.
- * Do not turn power on or turn power off with a diskette loaded in the drive. And, avoid placing diskettes in a location where they can be affected by a magnetic field.

SYSTEM REQUIREMENTS

As a minimum, the FODS requires a single disk drive, a disk interface or controller card and main memory in locations 6000 to 7FFF. The disk controller card is addressed to location A880 in OMNIDISK 65/5 series systems and to A8A0 for the OMNIDISK 65/8 series. Zero page locations from A0 to AF are reserved for FODS. The FODS operating system is resident in locations 7000 to 7FFF. Any data written to this memory block will cause a failure of the system with unpredictable results.

User memory (RAM or ROM) or interface circuits for peripheral devices should not be addressed to locations A800 thru A900. This will avoid conflict with the disk controller. HDE cannot, on an individual user basis, reassemble the system to accomodate such conflicts.

LOADING, INITIALIZATION AND ENTRY

The HDE FODS is self initializing and is loaded using the bootstrap program provided in this manual.

A correct load and initialization is indicated by the issue of the FODS sign-on and copyright message.

DISK INDEX

The disk index is a 512 character file written to track 0, sectors 1 through 4. The memory location used for disk index transactions is 7000 to 71FF. Because of this, user programs located in this region will be overwritten by the index during any disk transactions and files cannot be written to or saved from this area of memory.

A maximum of 40 files may be written to the disk. This includes deleted and active files. A status error will be issued when the 41st file save is attempted.

FILE CONCEPT

Files are opened by loading a file to memory from indexed disk. Scratch pad loads to memory do not open files, since the usual operating mode is to use scratch pads as temporary storage while working in a file. A trap has been included to inhibit the use of some system commands until a file has been opened with the editor. Commands operative without an opened file are; LOD, DEL, DIR, RUN and routines called from the system disk (FRE, LDN, etc.)

Any number of files may be in memory concurrently. Transfer among files is accomplished with the Text Editor 'FIL' command. If a file is transferred from indexed disk to memory using a 'LOD' command, that file is automatically opened and any subsequent edit or disk operations will be with that file.

A file may be of any length up to 32K characters, limited by contiguous memory.

DISK OPERATIONS - GENERAL

Files are written to disk either in indexed (all systems) or non-indexed form (OMNIDISK 65/8). Indexed files are written sequentially from track one thru 34 (66 for OMNIDISK 65/8). The 65/8 series write non-indexed files to two areas, A and B, from track 67 thru track 76. Saves to these locations will overwrite data previously recorded.

A file is written to indexed disk using the 'ENT' command and a file name. The name is entered into the disk index along with the parameters (starting/ending address, track and sector) for the file. A file written to indexed disk is 'protected' in the sense that subsequent saves will not overwrite any portion of the area in which the file is recorded. Any 'ENT' attempted using the same file name will result in the query 'ERASE FILE?' which is an indication to the user that a file with the same name has been located in the index. No further action will be taken to save the file without user intervention since a save will result in deletion of the old file.

Non-indexed disk file operations are referred to as 'scratch pad' because of their intended temporary nature. The A and B

scratch pads are not protected and each save to a particular scratch pad will overwrite previously stored data. Data in either 'A' or 'B' may be loaded to any valid memory location provided sufficient contiguous memory exists to contain the data. Loads to memory from scratch pad do not open files in the manner of indexed disk loads.

In 65/5 systems, the disk drive is normally off and turned on during a disk transaction. The drive is returned to off when the transaction is finished to keep disk wear to a minimum. See the 'COMMAND STRUCTURE' section for use of the '+' directive to turn drives on for disk loading.

DISK OPERATIONS - SINGLE DRIVE SYSTEMS

In single drive operations, several command routines will pause and output the prompt TO CONTINUE - ENTER "RETURN" This pause allows the system disk to be removed and the target disk to be inserted before the commanded action is resumed.

DISK OPERATIONS - DUAL DRIVE SYSTEMS

In dual drive systems, drives 1 and 2 are designated the 'system' drive and 'applications' drive respectively. All 'ENT' and 'LOD' commands are directed (by default) to the applications drive (#2). 'RUN' access to the system drive (#1) is made using the command structure. In 65/8 systems, the scratch pads may be used on either the system or applications drives.

The system defaults to the applications or systems drive, as appropriate. The disk transaction may be directed to other than the default disk by inserting the disk number in the operand as in the following examples:

```
ENT 1/$3000$3FFF=%MYFIL
ENT 1/SRCFIL
LOD 1/$2000=%MYFIL
LOD 1/SRCFIL
NAM 1/SRCFIL=ABCFIL
```


BOOTSTRAP ROUTINE

The FODS is loaded to memory with the bootstrap program provided in Appendix 1. The bootstrap may be loaded on cassette tape or paper tape depending upon the needs of the user. It is suggested that a copy of the bootstrap object code be made prior to attempting to load the system, since the program is self modifying.

The bootstrap is executed at location hex 0000. A display of the statement "FODS, COPYRIGHT 19XX, HDE, INC" and a FODS prompt character (#) indicate the system is loaded and ready for use.

After the bootstrap is loaded, insert the system disk into the drive unit and run the bootstrap by setting address 0000 with the SYM monitor 'G 0 <ret>' command. The system will be loaded in less than three seconds. If you do not receive the statement "FODS, COPYRIGHT 19XX, HDE, INC." followed by the FODS prompt '#', you may have an error condition. Recheck the following before a fault with the system is assumed:

- * Is the disk inserted properly into the drive unit?

If the disk drive is placed horizontally, the door should open upward. The disk label must face upward and be in the front.

- * Have all of the connections among the various components been made properly? Is the disk power supply on?

Please recheck all required interconnections and proper switch settings on the interface card before contacting HDE, inc.

- * If the trouble persists, contact HDE, inc. at the address found elsewhere in this manual, or telephone (201) 362-6574 for information about returning your system for replacement or repair. Do not forward your system to HDE before receiving instructions on the procedure to be followed.

FODS USER MANUAL - SYM VERSION

SYSTEM COPIES

The FODS will copy itself to disk and this may be accomplished at any time. FODS must be the FIRST file on the disk if it is to be loaded with the bootstrap routine provided in this manual. To copy the system use the command:

```
SAV $7300$7FFF=(version name)
```

To insure that all cold start initialization values are retained, DO NOT call 'TED' prior to saving FODS.

FODS ENTRY

The cold start (bootstrap) entry for FODS is hex 7300. Upon cold start, several locations for disk and supporting routines are initialized, the HDE copyright notice is output, the drive(s) are turned off (if applicable) and the SYM IRQ vector is set for FODS operations. At the completion of these steps, FODS is ready for operation.

FODS will reset the IRQ vector each time it is entered via warm start at 7380. Because of this, user installed routines may make an orderly return to FODS by terminating with a 'ERK' instruction (hex 00). This will insure that the 65/5 system drives are turned off.

After bootstrap, FODS entry from SUPERMON must be via the warm start location, 7380.

Entry to FODS at any other location is not recommended.

ZERO PAGE USAGE

FODS shares zero page locations A0 to AF with BASIC on a non-interference basis.

RAE OPERATIONS

The SYM Resident Assembler Editor (RAE) is linked to FODS by entering 'RAE <cr>' as a FODS command. The source file save and load functions are performed in RAE by entering 'ENTER filename' and 'LOAD filename' respectively. (Only the first two characters need be provided, e.g. EN or LO. Thus, LOAD FILE1, LO FILE1 and LOST FILE1 will all be processed the same.) An ERROR 30 will occur if the filename is not provided in both commands. To use some other disk function, enter 'DC commandstring'. For example, to display a disk directory enter 'DC DIR 1'. An ERROR 32 will occur if a commandstring is not provided. RAE source files are identified by a colon (:) as the first character of the file name, therefore, names should be limited to a maximum of 5 characters.

To append a file on disk to a RAE resident file, a plus (+) must preface the filename in the LOAD command.

ERROR NOTIFICATIONS

General

- ERROR The command interpreter does not recognize the entry as a legal command.
- INDEX UNDEF The current disk does not have an index.
- UNDEF The file name called cannot be found in the index.
- FILE? A file has not been opened and an 'ENT' or other file dependent command has been attempted. If this error persists return to the monitor and check location 00A9 for a zero value. FODS will not permit file operations with zero page addresses. If this location value is 00, any non-zero value will restore proper system operation.
- A file load activity (LOD, LSA, LSB) has resulted in an overflow from available memory.

Disk Status Errors - OMNIDISK 65 systems

- 11 or 51 Fatal error. Timing out fault. The disk command did not complete executing within the programmed time limit (generally 20 sec.). Indicates a probable hardware failure. If the error persists, record the values in locations \$7210 to \$721D to assist HDE in diagnosing the problem.
- 12 Fatal error. Unexpected interrupt. The disk controller interrupt flag register contained some flags set that were not expected at this time. Indicates a probable hardware failure. If the error persists, record the values in locations \$7210 to \$721D to assist HDE in diagnosing the problem.
- 13 (Reserved)
- 14 Fatal error. Format error. One of the following conditions occurred during an attempt to format a diskette:
- Drive not ready - most likely it was not turning or the diskette was not inserted.
- Diskette WRITE protected.
- Processor unable to achieve the required data rate. Be sure 6502 clock is at 1 mhz.
- 15 Fatal error. Step error. A step command was not properly executed during an attempt to format a diskette. Most probably a hardware failure. The specific conditions causing the failure are
- Drive not ready. It had to be ready previously to get to this point.
- Drive WRITE protected. It had been unprotected previously to get to this point.
- Controller has detected a seek or CRC error. This test is not made at this point.
- Drive failed to move off track 00.

- 16 Fatal error. Range fault. The table area used for formatting and checking a diskette has wrapped around memory. The most probable cause is a faulty processor.
- 17 Fatal error. Illegal disk command. If this error occurs record the values in locations \$7210 to \$721D to assist HDE in diagnosing the problem.
- 18 (Reserved)
- 19 (Reserved)
- 1A Fatal error. Seek error. Controller was unable to locate the desired track number at the current head location or format data on the diskette contains a CRC error. Most likely cause is a bad or marginal diskette. If the problem is associated with a particular diskette, copy the data to another diskette and reformat.
- 1B (Reserved)
- 1C (Reserved)
- 1D Fatal error. Irrecoverable data. The system has gone through 30 attempts to perform the transfer requested. Most probable causes are disk write protected, a bad diskette or misaligned drive.
- 1E (Reserved)
- 1F Fatal error. No controller. The controller requested does not exist or is not initialized. Probable causes are:
- You have requested to format using a controller that does not exist your system.
- The reset signal has occurred and you have reentered the system at other than the legal starting address.

FODS USER MANUAL - SYM VERSION

- 20 Fatal error. No drive. The drive does not appear in the system. Probable causes are:
- You have requested to format using a nonexistent drive.
- The drive is not rotating the disk
- 21 Fatal error. No diskette.
- 22 (Reserved)
- 23 Fatal error. Compare fault. Read after read or read after write check failed.
- Read after read check may fail if you have bad memory or attempted a load to nonexistent memory.
- A read after write check failure is most likely the result of a transmission error in the processor bus.
- 24 Fatal error. Closure error. The number of bytes transferred during a format operation was out of tolerance, most probably caused by out of tolerance disk speed.

Disk Status Errors - ALL SYSTEMS

- 28 See error #40
- 30 The file is too big. The maximum file size is 32,767 bytes (decimal). If the 'ENT' or 'LOD' attempt was made from RAE, check to see that the filename was specified.
- 31 A warning that the file loaded exceeds the RAE file boundry limits. Reset the the limits before continuing.
- 32 A commandstring was not included as part of a 'DC' command from RAE.
- 40 The current save will result in more than 40 files in the index. The save has been aborted.
- 50 The current save will result in a write to track 67 (scratch pad A). The save has aborted.
- In 65/5 systems this indicates that the current save will result in a write beyond track 33, the last track on the disk. The save has aborted.
- 60 A save has been attempted and a file is undefined, or a load has been attempted to zero page. See 'FILE?' error.
- 99 Starting sector address is 0. This is a fatal error and no additional saves should be attempted. Active files must be moved to another disk and the index deleted.
- FF The ending sector has been computed to be zero or greater than 16 or 26. If the error persists, the disk index must be deleted. See error 99.

COMMAND STRUCTURE

Each FODS command is based upon a three character mnemonic which attempts to describe to some extent the function involved. Commands must be entered as shown in each command description. Errors are answered with 'ERROR'.

The user may add functions by saving the object level routine to the system disk and using a three character file label (%aaa) that does not conflict with existing commands. The user routine may then be called as any other FODS command. Be sure that the routine starts with an executable instruction.

Several directives are available depending upon the system. These are;

- ^C - 'control C' is used to exit FODS and return to SUPERMON.
- + - The '+' sign is used in 65/5 systems to start the drives for proper disk alignment and loading. Insertion of a disk into a drive that is not rotating may cause jamming of the drive and possibly damage the disk. Since it is not possible to start the drive rotating before bootstrap, insure that the disk is centered in the protective jacket prior to placing it in the drive. If the door appears to bind on closing, remove the disk and recenter it.

COMMAND SUMMARY

*FM	(FORMAT DISK)
BAS	(RUN BASIC)
CPY <n>	(COPY TO DISK n)
CPY <n>A	(COPY THE ACTIVE FILES TO DISK n)
CPY <n>/<name1>,.<namen>	(COPY THE named FILES TO DISK n)
DEL INDEX!	(DELETE INDEX!)
DEL <name>	(DELETE FILE name)
DIR <n>	(DISPLAY DIRECTORY ON DRIVE n)
ENT <name>	(SAVE FILE name)
ENT <\$hhhh\$hhhh=name>	(SAVE FILE FROM hhhh TO hhhh WITH name)
FRE <n>	(BYTES REMAINING ON DISK n)
LDN <n>	(LOAD FILE n)
LOD <name>	(LOAD FILE name)
LOD <\$hhhh=name>	(LOAD FILE name TO HEX ADDRESS hhhh)
LSA	(LOAD SCRATCH PAD A)
LSB	(LOAD SCRATCH PAD B)
LSA <\$hhhh>	(LOAD SCRATCH A TO HEX ADDRESS hhhh)
LSB <\$hhhh>	(LOAD SCRATCH B TO HEX ADDRESS hhhh)
NAM <old name>=<new name>	(CHANGE old name TO new name)
NAM <file #>=<new name>	(CHANGE file # name TO new name)
PAK <n>	(PACK DISK n)
REA <name=\$hhhh>	(CHANGE ADDRESS OF name TO HEX hhhh)
RUN <name>	(RUN OBJECT PROGRAM name)
SOR <col#><A or D>	(SORT column # Ascending or Descending)
SSA	(SAVE FILE TO S/PAD A)
SSB	(SAVE FILE TO S/PAD B)

*FM (FORMAT DISK)

OMNIDISK 65 systems use "soft sectored" diskettes which must have written to them the proper track and sector address information before the system can access the diskette. This command is used to load and initiate the disk formatter.

When called, the following prompts are issued:

ENTER:

DRIVE NUMBER? : (enter 1 or 2 - as appropriate)

"+" TO FORMAT : (insert the diskette to be
formatted into the selected
drive, then enter "+".)

Insure that you are not attempting to format the system disk or a disk that contains data that you wish to retain since the format routine will overwrite the entire disk. Any data recorded will be nonrecoverable. HDE WILL CHARGE THE CURRENT RATE FOR REPLACEMENT OF ANY SYSTEM DISK OVERWRITTEN WITH THIS ROUTINE. The user must carefully ascertain that the system disk has been removed or that the proper drive is selected before answering the last prompt with "+" and entering the "RETURN".

RESTRICTIONS: See above.

EXAMPLE: *FM

DISCUSSION:

Mini-diskettes (5") must be formatted with this routine prior to first use. Full size (8") diskettes are usually pre-formatted to the IBM single density standard (77 tracks, 26 sectors/track and 128 bytes/sector). Full size diskettes complying to this standard will operate without the need to pre-format.

The disk format routine must have the ability to format any soft sectored diskette regardless of the absence or presence of formatting information. As a result there are no checks made regarding the validity of the index (if one has been recorded to the diskette). The user must be absolutely sure that the action prescribed in answers to the prompts define the desired action. THE *FM ROUTINE CAN OVERWRITE THE SYSTEM DISK. IF THIS OCCURS THE PROGRAMS CANNOT BE RECOVERED AND MUST BE REPLACED.

BAS

(RUN BASIC)

This is the call for the disk interface routines to SYM BASIC. The call must be used to properly initialize the appropriate pointers and link the disk facilities to BASIC. All of the standard features of BASIC are available with the addition of the directives and commands described below.

BASIC programs developed with the disk interface installed will be originated at location \$0201. Program files are address independent. That is, the loading address (as shown in the index when the FODS 'DIR' command is used), may be changed (with the FODS 'REA' command) and the program will operate at the new location. If this technique is used, insure that at least one RAM byte is available at the address immediately ahead of the new address. For example, if address 3FFF in your system is a ROM address, a BASIC program will not operate if it is loaded to 4000, but will operate if it is loaded to 4001 (assuming there is sufficient memory starting at these locations to contain the program).

All disk commands initiated from BASIC are preceded by a period. Commands may be used in the command (direct) or run modes. There are no restrictions in the use of the commands other than those imposed by BASIC. File names should be enclosed with quotes ("name") and are limited to five characters in length. FODS will insert a "@" sign in the file name to identify it as a BASIC file. In dual drive configurations, actions will default to the applications drive unless the system drive is specified by inserting a 1 following the appropriate command. For example, the command .EN1,"NAME" will save the current program to the system drive.

File names may be expressed as string variables (SAVEP,A\$), provided that the variable has been previously defined.

The trigonometric functions for SIN, COS, TAN and ARCTAN are included within the package. For example, the function SIN(X) will return the sine of the expression X with the value of X interpreted as being in radians. Note that 1 radian = 180/PI degrees = 57.2958 degrees. Therefore the sine of X degrees is given by the expression SIN(X/57.2958).

Several other functions have been included:

- ^C (control C) Exit to SUPERMON. To reenter BASIC simply type G<cr>.
- ^H (control H) Backspaces and deletes one character per entry.
- ^X (control X) Cancels the current line.

FODS USER MANUAL - SYM VERSION

`^A` (control A) Initiates (cancels) AUTO-LINE NUMBERING. Upon initial entry of `^A`, BAS will echo an 'A' to the terminal. None, one or two arguments may then be included. If no arguments are supplied, AUTO-LINE NUMBERING will start at line number 10 and increment subsequent line numbers by 10. Entering one argument will specify that value as the starting line number and the increment will default to 10. Entering the second argument will define that value as the increment. To cancel AUTO-LINE NUMBERING, enter `^A`. Lines in the file may be skipped by entering `^X` or deleted by entering a `<cr>` immediately following the number.

Warm start entry for 8" systems is \$5006 and for 5" systems is \$6009.

All other functions are accessed by a three character command as described in the following list.

FODS USER MANUAL - SYM VERSION

The following commands are available;

- .DI Display the disk directory.
- .LO,"name" Load the program "name"
- .CH,"name" Load and run "name" preserving current data files (chaining). Insure that each program chained is smaller than the program that originally created the data files.
- .AP,"name" Append file "name" to the program in memory. All line numbers and line references for GOSUB, GOTO, etc. statements in the appended file will be resequenced appropriately.
- .RU,"name" Load and run the file "name"
- .LO%, "name" Load the object file "name". This is a convenient method for loading programs that subsequently will operate with 'USR', etc.
- .LO#, "name" Load the program data file "name". Data files are address independent and will be appended to the end of the program file in memory.
- .EN,"name" Save the program file with "name"
- .EN#, "name" Save the program data file with "name" String variables defined in a program (A\$="ABCDEFGH") must be concatenated if they are to be included as a part of the saved data file (A\$="ABCDEFGH"+"")
- .DC,"<FODS command>" Execute the FODS command specified. For example, .DC,"FRE 2" will call and operate the FODS command 'FRE'. Care must be taken to avoid using a routine that will overwrite the BASIC interface routine. DO NOT USE '*FM' OR 'CPY' WHILE IN BASIC.
- .BY Exits BASIC and returns to FODS.

.NU<bln,incr,lms,lne>

This routine is loaded from disk and is used to renumber BASIC programs. The parameters are defined as follows:

- bln = The beginning line number in the resequenced program.
- incr = The increment between consecutive lines.
- lms = The (old) line number to begin renumbering at.
- lne = The (old) line number to halt renumbering at.

All parameters are optional. If not supplied, the default values are: 100,10,0,32767.

.GT<expression>

A computed GOTO. For example, if Y = 4, then .GT Y*Y-3 will cause a jump to line 13.

RESTRICTIONS: None

EXAMPLE: BAS

DISCUSSION:

'%BAS' includes a default 'size memory' of 8192 to avoid overwriting the disk routines in the event the system contains contiguous memory from \$0200 to \$7FFF. This default is supplied if the <cr> is used in response to the memory size question upon initialization. To insure that all desired memory is included, respond with the decimal value of the highest memory location that you wish to use. In any event, do not enter a value greater than 24,576. In addition, 'BAS' will deduct one page (256 bytes) from the specified memory size to preserve appropriate pointers for data saves and loads. If you do not have memory to \$2000 (8192), use of the <cr> default will result in a loss of data.

```

CPY <n>                (COPY TO DISK n)
CPY <n>A              (COPY THE ACTIVE FILES TO DISK n)
CPY <n>/<name1>,<namen> (COPY THE named FILES TO DISK n)

```

The 'CPY' command is provided with dual drive systems only. In the first case, the command 'CPY <n>' will copy all files from the other drive to the drive specified by <n>. If <n> is followed by 'A' only the active files will be copied. Named files may be copied as shown in the third case. Each name must be followed by a comma except for the last one. As many names may be entered as will fit on one line.

RESTRICTIONS:

'CPY' operates by loading each file called to memory and then saving the file from memory to the specified disk. If an active file exists in memory that must be retained, that file should be saved to disk prior to use of this function.

```

EXAMPLE: CPY 1
          CPY 2A
          CPY 2/%FODS3,%TED,%CPY

```

DISCUSSION:

A blank disk cannot be copied. If an attempt is made to copy a blank disk, the prompt "TARGET DISK?" will be output. The correct disk number should be entered at this time. Note that 'n' represents the target disk, not the disk being copied.

```

DEL INDEX!           (DELETE INDEX!)
DEL <name>           (DELETE FILE name)

```

Deletes the index of the applications disk. When the index is deleted, the files on the disk may not be recovered.

Deletes the file specified. When the file name is deleted the file can be recovered using the 'LDN' command.

RESTRICTIONS:

Because of the nature of this command, the complete string 'INDEX!' must be entered before it will be executed.

```

EXAMPLE: DEL INDEX!
          DEL FILEA

```

DISCUSSION: None

 DIR <n>

(DISPLAY DIRECTORY ON DRIVE n)

Causes the index of the current disk as specified by <n> to be read into memory and displayed on the output device.

RESTRICTIONS: None.

EXAMPLE: DIR 1

DISCUSSION:

The 'DIR' command may be used at any time to display the specified disk index. The file parameters are shown for both active and deleted files. The format is as follows:

Sequence Number	File Name	Starting Address *	Ending Address *
--------------------	--------------	-----------------------	---------------------

* In memory

The absence of a file name indicates a deleted file.

The return of a prompt without any other display or message indicates that data has been saved to a scratch pad.

The last line displayed shows the next track 'T' and sector 'S' to which the next file save will be made.

ENT <name> (SAVE FILE name)

ENT!<name> (SAVE FILE name)

This command is used to name a file and store it on disk. The file name and parameters are recorded in the disk index.

RESTRICTIONS:

If an 'ENT' is attempted with a file named the same as a file already on disk, the FODS will respond with "ERASE FILE?". If the operator responds with N, the save is aborted. A Y will cause the save to continue and the old file parameters will be deleted from the index.

A dollar sign (\$) may not be used as the first character of the file name.

A percent sign (%) as the first character identifies the file as an executable program.

A 'commercial at' sign (@) as the first character of a file identifies the file as a 'BASIC', 'FOCAL' or similiar file.

A 'colon' (:) identifies the file as a RAE source file.

EXAMPLE: ENT FILEAB
ENT!NEWFIL

DISCUSSION:

A file name may be any character string, except as limited above, up to a total of six characters. If more than six, the name will be truncated to six. If less than six, the FODS will pad the name with spaces to equal six.

The command form 'ENT!<FILE NAME>' is used when it is desired to inhibit the comment "ERASE FILE?". In this case, the old file will be deleted and the new file entered without a notification by the system, if a duplicate file is found in the disk index.

ENT <\$hhhh\$iiii=name> (SAVE hhhh TO iiii WITH name)

This command defines a file by starting address, ending address and name. It allows the save of otherwise undefined files or programs. If the file saved is an object program the first character of the name must be percent (%) for it to be called by the 'RUN' command.

RESTRICTIONS:

The first address must be the low order address of the file for proper operation.

Files located in zero page and page one (0000 to 01FF) cannot be saved with this command.

EXAMPLE: ENT \$2000\$3EFE=%UPDAT

DISCUSSION:

Note that a dollar sign (\$) must precede each address specification. Leading zeros need not be entered. The delimiters \$,\$ and = must not be bounded by spaces.

FRE <n> (BYTES REMAINING ON DISK n)

'FRE' will load the appropriate disk index and calculate the number of bytes of free storage remaining. The result is output in decimal and hexadecimal bytes.

RESTRICTIONS: None

EXAMPLE: FRE 2

DISCUSSION: None

LDN <n> (LOAD FILE n)

This command permits the loading of any indexed file on the applications disk by file number (the leftmost number in the index display), thereby allowing the retrieval of deleted files.

RESTRICTIONS: None

EXAMPLE: LDN 15

DISCUSSION: None

LOD <name> (LOAD FILE name)
 LOD <\$hhhh=name> (LOAD FILE name TO HEX hhhh)

This command causes the current disk index to be read into memory. The index will be searched for the specified file name and, if found, the corresponding file will be read into memory. The load location for the file is that location from which the file was originally stored to disk unless the <\$hhhh=name> option is used, in which case the file loads at the address specified. If the name is not found, the system will respond with 'UNDEFINED' indicating that the file name cannot be located. If the disk index has not been initiated, the system will respond with 'INDEX UNDEFINED'. If a name is less than six characters it does not need to be padded with spaces to make it equal to six.

RESTRICTIONS:

No attempt is made to distinguish between an object or data file when using the form 'LOD <hhhh=name>.

EXAMPLE: LOD FILEAB
 LOD \$5000=FILEAB

DISCUSSION: None

LSA	(LOAD SCRATCH PAD A)
LSB	(LOAD SCRATCH PAD B)
LSA <\$hhhh>	(LOAD SCRATCH A TO HEX ADDRESS hhhh)
LSB <\$hhhh>	(LOAD SCRATCH B TO HEX ADDRESS hhhh)

This command causes the data in the corresponding scratch pad (A or B) to be loaded into memory as specified by the operand. The default address (no operand) is the current file. If the hex address is specified, the selected scratch pad data will be loaded to that address. Note that the hex address must be preceded with a dollar sign (\$).

RESTRICTIONS:

For 65/8 systems only.

EXAMPLE: LSA (or LSA.)
 LSB \$4000 (or LSB.\$4000)

DISCUSSION:

In addition to being a temporary storage location, each of the scratch pads offer a means for rapidly relocating files in memory as well as linking different files. To move a file in memory, use the SSA or SSB command to save a file to one of the scratch pads. Then use the LSA or LSB command with a location for reloading the file (e.g., LSA \$3500). The file will now be loaded to the new location.

Although the scratch pads are considered temporary, the parameters are included in the disk index. Therefore, scratch pad files will be retained until they are overwritten although a disk is removed.

In dual drive systems, if a period is included immediately following the command, as in the examples above, the transaction is with the 'system' disk (drive #1).

NAM <old name>=<new name> (CHANGE old name TO new name)
 NAM <file #>=<new name> (CHANGE file # name TO new name)

Permits changing the name of a file in the disk index. Deleted files may be renamed by using the file number instead of the old file name.

RESTRICTIONS:

If an attempt is made to change the name of a file to one which already exists in the index the routine will reply with the prompt 'DUPLICATE NAME' and abort.

EXAMPLE: NAM OLDFIL=NEWFIL Reference by old file name
 NAM <7>=NEWNAM Reference by old file number

DISCUSSION: None

PAK <n> (PACK DISK n)

'PAK' will pack the specified disk by moving active files down into the areas occupied by deleted files. The routine offers a convenient means of compressing disk files to eliminate the area reserved for inactive files. 'PAK' may be used at any time without effect upon any files currently in memory. When invoked, the prompt "PACK: IN PROGRESS:" will be output. When the packing has been completed, "COMPLETE" will be displayed.

RESTRICTIONS:

Any disruption, such as power failure, computer RESET or opening of the disk drive door during the operation will cause loss of the files. File management practices should include procedures which will insure a backup file is available.

EXAMPLE: PAK 2

DISCUSSION: None

REA <name=\$hhhh>

(CHANGE ADDRESS OF name TO HEX hhhh)

Changes the address of the file "name" to the hex address specified by \$hhhh. If the file is not found in the index the prompt "ERROR" is displayed.

RESTRICTIONS: None

EXAMPLE: REA %MYFIL=\$4000

DISCUSSION:

The ending address is calculated and changed to correspond to the correct file length.

RUN <name>

(RUN OBJECT PROGRAM name)

Will search the applications disk for the file, load the file into memory and start executing the file at the first location. To use this command, a file must be a program with the first location an instruction.

RESTRICTIONS:

The file name must begin with a percent (%) sign.

EXAMPLE: RUN %UPDATE

DISCUSSION:

This command offers the user a convenient method for loading a program (in object form) into memory and executing the program.

SOR <col #><A or D> (SORT column # Asc or Desc)

'SOR' is the call for an alphanumeric sort routine for TED files. The records in a TED file may be sorted either in ascending or descending order starting with the column number specified.

RESTRICTIONS:

The routine initially scans the file to determine if any lines are out of range of the sort. For example, if column 55 is specified as the start column, all lines in the file that are less than 55 characters long will be displayed following the statement "OUT OF RANGE", and the sort is aborted.

EXAMPLE: SOR 55A
 SOR 55 (Ascending sort default)
 SOR 12D

DISCUSSION: None

SSA (SAVE FILE TO S/PAD A)
SSB (SAVE FILE TO S/PAD B)

This command causes the current file (as defined by the operand) to be saved to the respective scratch pad.

RESTRICTIONS:

For 65/8 systems only.

Each scratch pad is approximately 16k characters. A save to scratch A may be up to 32 characters. But, this will overwrite scratch B. A save to B greater than 16k will result in a loss of data.

EXAMPLE: SSB (OR SSB.)

DISCUSSION:

The scratch pads are not protected. Therefore, use of this command to save data will result in the loss of data previously written to the respective scratch pad.

In dual drive systems, if a period is included immediately following the command, as with the examples above, the transaction is with the system disk.

FODS BOOTSTRAP

The bootstrap must be keyed into zero page starting at hex address 0000. After verifying that the correct code has been entered, a tape copy should be made before attempting to load FODS, because the program is self modifying. If the program is loaded into EPROM a routine to move the bootstrap to page 0 must be provided.

FODS BOOTSTRAP VERSION 1.0 (OMNIDISK 65/5)

```

0000 20 A7 00 A9 BF 8D 82 A8 A2 00 A0 00 E6 C3 E6 C3 E8 D0 F9 C8
0014 D0 F6 86 C3 A9 73 85 C4 A0 F4 20 92 00 20 A0 00 D0 F6 A0 A4
0028 20 92 00 20 A0 00 D0 5C A2 FE CE 83 A8 A9 10 8D 80 A8 8E 81
003C A8 A0 77 20 92 00 EE 83 A8 A9 38 8D 80 A8 A0 7F AD 8D A8 29
0050 02 F0 F9 AD 81 A8 91 C3 D0 00 88 10 EF 20 A0 00 D0 2A D8 18
0064 A5 C3 69 80 85 C3 A5 C4 69 00 85 C4 C9 80 D0 11 A5 59 D0 17
0078 A9 D1 85 56 A9 32 85 59 A2 00 4C 16 00 E0 EF F0 9D CA D0 A6
008C 4C 03 80 4C 00 73 A9 FF 8D 83 A8 A9 00 8D 80 A8 8C 81 A8 60
00A0 AD 8D A8 29 10 F0 F9 A9 00 8D 83 A8 A9 20 8D 80 A8 A9 DB 8D
00B4 8C A8 A9 FB 8D 8C A8 AD 81 A8 49 FF 29 98 60

```

FODS BOOTSTRAP VERSION 1.0 (OMNIDISK 65/8)

```

0000 20 A9 00 A9 BF 8D A2 A8 A2 00 A0 00 E6 C5 E6 C5 E8 D0 F9 C8
0014 D0 F6 86 C5 A9 73 85 C6 A0 F5 20 94 00 20 A2 00 D0 F6 A0 A5
0028 20 94 00 20 A2 00 D0 5E A2 FE CE A3 A8 A9 10 8D A0 A8 8E A1
003C A8 A0 77 20 94 00 EE A3 A8 A9 38 8D A0 A8 A0 00 AD AD A8 29
0050 02 F0 F9 AD A1 A8 91 C5 D0 00 C8 C0 80 D0 ED 20 A2 00 D0 2A
0064 D8 18 A5 C5 69 80 85 C5 A5 C6 69 00 85 C6 C9 80 D0 11 A5 59
0078 D0 17 A9 D1 85 56 A9 34 85 59 A2 00 4C 16 00 E0 E5 F0 9B CA
008C D0 A4 4C 03 80 4C 00 73 A9 FF 8D A3 A8 A9 00 8D A0 A8 8C A1
00A0 A8 60 AD AD A8 29 10 F0 F9 A9 00 8D A3 A8 A9 20 8D A0 A8 A9
00B4 DB 8D AC A8 A9 FB 8D AC A8 AD A1 A8 49 FF 29 98 60

```