TEKTRONIX
Information Display Group
P.O. Box 500
Beaverton, OR 97077
Telephone (503) 682-3411

UNITED STATES
FIELD OFFICES

ALABAMA
Huntsville

ARIZONA
Phoenix

CALIFORNIA
Concord
Irvine
San Diego
Santa Clara
Woodland Hills

COLORADO
Denver

CONNECTICUT
Milford

FLORIDA
Fort Lauderdale
Orlando
Pensacola

GEORGIA
Atlanta

ILLINOIS
Chicago

INDIANA
Indianapolis

KANSAS
Kansas City

LOUISIANA
New Orleans

MARYLAND
Baltimore
Rockville

MASSACHUSETTS
Boston

MICHIGAN
Detroit

MINNESOTA
St. Paul

MISSOURI
St. Louis

NEW JERSEY
Springfield

NEW MEXICO
Albuquerque

NEW YORK
Albany
Long Island
Poughkeepsie
Rochester
Syracuse

NORTH CAROLINA
Raleigh

OHIO
Cleveland
Dayton

OKLAHOMA
Oklahoma City

OREGON
Portland

PENNSYLVANIA
Philadelphia
Pittsburgh

TEXAS
Dallas
Houston
San Antonio

UTAH
Salt Lake City

VIRGINIA
Hampton

WASHINGTON
Seattle

INTERNATIONAL
FIELD OFFICES

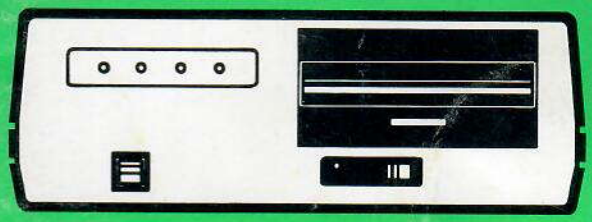
AUSTRALIA
Sydney

CANADA
Montreal

HOLLAND
Badhoevedorp

JAPAN
Tokyo

REV B, APR 1979



4907 File Manager

Operator's Pocket Reference

Tektronix®
COMMITTED TO EXCELLENCE

INTRODUCTION

This 4907 Pocket Reference includes:

- FILE MANAGEMENT INITIAL STEPS
- POSITION AND SIZE OF STATUS MESSAGE FIELDS
- HOW LARGE SHOULD A FILE OR FILE RECORD BE?
- HOW TO WRITE A FILE IDENTIFIER
- SPECIAL CHARACTERS IN FILE IDENTIFIERS
- FILE ATTRIBUTES
- COMMAND EXAMPLES

FILE MANAGEMENT INITIAL STEPS

The following steps will prepare the 4907 for file management operations:

1. Turn system power on and place a flexible disc, label up, in the drive.

2. Type **INIT**

   This initializes the system and sets all variables to an undefined state.

3. Type **CALL"SETTIM","DD-MON-YY HH:MM"**

   This command, with the current date and time, sets the system clock.

4. Type **DIM A$(300)**

   This command dimensions A$ to 300 bytes to allow room for status messages.

5. Type **CALL"MOUNT",0,A$**

   This command tells the system that there is a formatted disc in device 0 that is ready for use. If the disc is blank and must be formatted, a CALL"FORMAT" will prepare the disc for use and execute an automatic CALL"MOUNT". A CALL"DRES" which reserves the device (drive) is necessary before the disc can be formatted. A CALL"DREL" is necessary after formatting to release the device.

6. Type **A$**

   By typing this string variable, you can review the status message generated by the previous CALL"MOUNT". A typical device status message follows:

```
4907     DEV ID     INCTAX     VOL ID
  SMITH & CO.     OWNER
    250000 FREE     630784 SIZE        0 LOST
  256 BLK SIZE
07-MAR-77 10:30 FORMATTED     0 FILES OPEN
```

   This message is generated for a device identified as the 4907. The disc is identified as INCTAX and the owner as SMITH & CO. There are 250,000 bytes of usable nonfile space. The 630,784 size field indicates this is a double density flexible disc. The 0 LOST message indicates that all free space is accessible. The block size is 256 bytes.

   This disc was last formatted on March 7, 1977 at 10:30 AM. No files on the disc are open. The device is not reserved (RESERVED does not appear in the status message) or write protected.

   The unit number appears at the beginning of the message when CALL"CUSTAT" instead of CALL"MOUNT" is executed.

7. If any files are on the disc, then complete status messages of all files may be displayed by typing **DIR2,"@"**. A typical complete file status message follows:

```
UNIVERSITY/RECORD
BRS N ATR     30208 ALLOC     12-DEC-77 08:30 ALT
    FILES10000 USED       12-DEC-77 10:30 USED
0 OPEN      200 REC LEN   12-DEC-77 08:30 CREATED
```

   This sample shows that a file with the F.I.(File Identifier) UNIVERSITY/RECORD was created December 12, 1977 at 8:30 AM and last used December 12, 1977, at 10:30 AM. This message shows the file has never been altered because the ALT field is exactly the same as the CREATED field. The file's ATtRibutes, BRSN, show the file is binary, private, scattered and noncompressible. This file has been allocated 30208 bytes of space, divided into records 200 bytes long, with 10000 bytes already used.

8. Files may now be created, written to, or read. The GENERAL SEQUENCE FLOW CHART in the 4907 Operator's Manual shows all the steps from start-up through system deactivation.

POSITION AND SIZE OF STATUS MESSAGE FIELDS

If your program needs device or file status message information, then the SEG function is useful. The following tables show the location in number of characters from the beginning of message and the length (number of characters) of each field.

Device Status Message

(186 characters long; generated by CALL"MOUNT")

Field	Starting Position/Length
DEV ID	1/10
VOL ID	23/10
OWNER	43/24
FREE	74/9
SIZE	91/9
LOST	108/9
BLK SIZE	122/8
FORMATTED	140/15
OPEN	169/3

If the device is RESERVED and WRITE PROTECTED, the device status message increases to 212 characters. These additional fields begin at position 186.

File Status Message

(189 characters long plus the characters in the File Identifier, which may be 59 characters maximum)

Field	Starting Position/Length
ATR	1/13
ALLOC	22/9
USED	83/9
REC LEN	145/9
OPEN	133/3
Date: ALT	42/15
USED	103/15
CREATED	165/15
FILE IDENTIFIER	189/59

Time Status Message

(18 characters long; generated by the CALL"TIME" command)

Field	Starting Position/Length
DAY	1/2
MON	4/3
YR	8/2
HOURS	11/2
MINUTES	14/2
SECONDS	17/2

HOW LARGE SHOULD A FILE OR FILE RECORD BE?

The number of bytes allocated for a particular file or file record depends on whether the file is ASCII or binary.

The following table shows how many bytes are required for data stored in files created using WRITE (Binary) or PRINT (ASCII).

	PRINT (ASCII)
Numeric values and strings	1 byte per character + 1 byte for CR

	WRITE (BINARY)
Numeric item	9 bytes for each numeric item
String item	4 bytes + 1 byte for each character
End of record item	1 byte ; random access file only

HOW TO WRITE A FILE IDENTIFIER

The File Identifier (F.I.) is placed in a command to tell the system where it can find an existing file or where to create a new file. The F.I. also assigns names to as many as four higher level libraries and to the file itself. It is also used to assign passwords and the file extension.

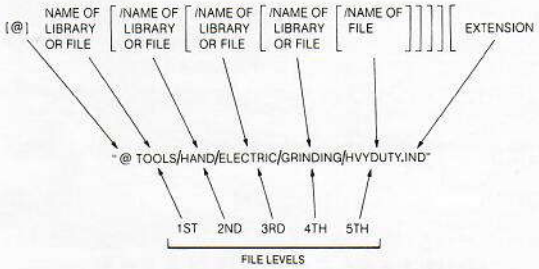
An F.I. can be as small as this example, representing a file on the 1st level only:

"TOM"

An F.I. can be as large as this one, representing five levels, each with its own password:

"FEDERAL:RED/STATE:BLUE/COUNTY:GREEN/MUNICIPAL:
YELLOW/LOCAL.DAT:BLACK"

The descriptive format for an F.I. and its relationship to a typical F.I. is shown in the following example:



FIELD DEFINITIONS

@	The commercial "at" sign allows accessing a library other than the current library specified with CALL"USERLIB". In the previous example, the system accesses the library TOOLS regardless of the current library.
TOOLS	This is the name of the 1st level library.
HAND	This is the name of the 2nd level library.
ELECTRIC	This is the name of the 3rd level library.
GRINDING	This is the name of the 4th level library.
HVYDUTY	This is the name of the 5th level file.

Each of these five fields may contain up to 10 characters. The first character must be alphabetic; the remaining characters are alphanumeric.

.IND	This is the extension allowing the system to distinguish between this and some other file named HVYDUTY in these same libraries. Only a file may have an extension. The extension consists of up to four characters, with the first being alphabetic.

Up to 21 characters of the F.I. can be placed in a CALL"USERLIB" command. These characters are then considered the "current library" and do not need to be repeated in subsequent commands. For example, the command CALL"USERLIB","A" enables one to reference the file A/B/C later as simply B/C.

PASSWORDS

A password may be added to any file or library name when created, to prevent unauthorized access. The file or library name is followed by a colon, followed by the password. For example:

TOOLS:PASSWRD

The password can be up to ten characters long; the first character must be alphabetic.

DELIMITERS

In a file identifier, a slash (/) separates name fields:

TOM/FRED

The entire F.I. must be contained in quotation marks:

"TOM/FRED"

A colon (:) separates the password from the name of the library or file:

"TOM/FRED:PASSWRD"

A period (.) separates the extension from the file name or its password:

"TOM/FRED:PASSWRD.EXT"

SPECIAL CHARACTERS IN FILE IDENTIFIERS

@ circumvents the current library that is defined by CALL"USERLIB" or by default.

e.g. "@B/DOG"
accesses file DOG in library B regardless of current library.

selects all eligible files at the level it is entered, as well as all subsequent levels.

e.g. "@B#"
accesses all libraries and files within library B.

* selects:

a. all eligible files at the level or levels the character is entered.

e.g. "@B/DOG/*"
selects all files in libary B/DOG.

b. all eligible files with a particular prefix.

e.g. "@B/DOG/CA * "
selects all files beginning with CA in library B/DOG.

c. all eligible files using extensions with a particular prefix.

e.g. "@B/DOG/MOUSE. * "
selects all MOUSE files with any extension in library B/DOG.

? selects all eligible files with identifiers that have any character in the ? position but match all other characters in the specified F.I..

e.g. "@B/DOG/?OUSE"
would select files such as MOUSE, LOUSE or HOUSE in library B/DOG

$ replaces @SYSLIB in the F.I. when addressing the SYSTEM LIBRARY.

e.g. "$GRAPH.PLT" equals "@SYSLIB/GRAPH.PLT"

FILE ATTRIBUTES

B	BINARY (default)
A	ASCII
H	HOST BINARY
R	PRIVATE (default)
U	PUBLIC
C	CONTIGUOUS
S	SCATTERABLE (default)
M	COMPRESSIBLE
N	NOT COMPRESSIBLE (default)

NOTE

The attribute SC (scatterable but currently contiguous) may be returned in a file status message. This attribute cannot be assigned by a CREATE or ASSIGN command.

COMMAND EXAMPLES

The name and general syntax of each command is shown, followed by one or two examples. The letters lfn mean logical file number as assigned by OPEN. Syntax information enclosed with square brackets is optional.

APPEND F.I.[,"ASCII"]; target line number [,increment between line numbers]

APP"@MYLIBRY/DAT";1150,15

Takes the entire program in file DAT in library MYLIBRY and places it in memory, starting at and replacing line 1150 of the current program. Line numbers increment by 15. The MEM function should be executed before APPEND to ensure sufficient space exists in memory.

APP"@SYSLIB/EDITOR","A";1000

Appends ASCII program starting at line 1000 using the default increment of 10.

ASSIGN F.I.;attributes (R,U,S,C,N,M)

ASS"@YRLIBRY/MATH";"R"

Changes the MATH file under library YRLIBRY to a private status. Refer to file attribute list for attribute descriptions.

CLOSE[logical file number]

CLO

Closes all files.

CLO 3

Closes the file identified as lfn 3.

INIT may be required to close all files after aborting an APPEND, OLD, or SAVE operation.

CALL"COMPRS", device address, compress control

CALL"COMPRS",2,1

Collects nonfile space as well as space from files not

containing information on device 2. Groups this space into larger contiguous blocks. Used only when disk is too full to create a needed contiguous file.

CALL"COMPRS",2,0

Collects only nonfile space on device 2 into larger contiguous blocks.

COPY...F.I. # 1, source device address TO F.I. # 2, target device address

COP"@X/Y/Z",1 TO "@A/Y/Z",2

Copies file Z from library X/Y on device 1 to library A/Y on device 2. If the new libraries and file do not exist, they are automatically created.

COP A$,1 TO B$,2

Copies information in the file named in A$ on device 1 to the file named in B$ on device 2.

CREATE F.I.[,attributes]; number of logical records, record length

CRE"@MYLIBRY/DATA","A";100,70

Creates the random access file DATA on the library MYLIBRY. The information is stored in ASCII in 100 records of 70 bytes each. Record contents in random access files must be initialized before use.

CRE"@MYLIBRY/C";10000,0

Creates a sequential access file of 10,000 bytes of information in the default binary format.

CALL"CUSTAT", target string variable

CAL"CUSTAT",A$

Generates a status message for all devices interfaced to the controller and stores the message in the specified string variable.

DELETE ALL

Erases the Graphic System's memory, closes all files, sets the current device to 0 and the current library to SCRATCHLIB.

DIRECTORY[I/O address:][format code[,F.I.]]

Format codes for directory lists:

0	File names only.
1	File names and dates.
2	Complete file status.

DIR2,"@MYLIBRY/GRADES"

Locates the file GRADES in library MYLIBRY; the 2 causes DIR to display a complete file status message on the Graphic System's screen. The screen is the default address of the message destination.

DIR or DIR0

Lists names (F.I.s) of all files on the current library.

CALL"DISMOUNT", device address

CAL"DISMOUNT",1

Deactivates device 1. Prevents opening any files on that device until another CALL "MOUNT" is executed.

CALL"DREL", device address

CAL"DREL",2

Releases reservation of device 2.

CALL"DRES", device address

CAL"DRES",2

Reserves device 2.

CALL"DSTAT", device address, target string variable

CAL"DSTAT",2,A$

Generates a status message about device 2, as well as a status message on all open files. This message is stored in A$.

CALL"DUP", source device address, target device address, compress control

CAL"DUP",3,2,1

Duplicates all information from device 3 to device 2. All space, including all allocated but unused file space, is collected into a contiguous space. (If the last field contains a 0, then unused allocated space is not collected.)

END

Stops program execution and closes files.

CALL"FFRMT", device address, volume I.D., 1, 1, owner I.D., master password, 1st, 2nd, 3rd, 4th, and 5th level chains

CAL"FFRMT",1,"LAB",1,1,"GENETICS","BLUE",1,10,1,1,1

Operates the same as CALL "FORMAT", except CALL"FFRMT" does not check the disc for bad blocks. This command is for quickly reformatting discs.

CALL"FILE", device address, F.I., target string variable

CAL"FILE",2,"@MYLIBRY/MATH",A$

Generates a complete status message about file MATH in library MYLIBRY on device 2. The message is stored in A$. If the file does not exist, the function LEN A$ would return a value of zero.

CALL"FMVALS", target numeric variable, target string variable

CAL"FMVALS",A,A$

Sends the current device address to A and the name of the current library to A$.

CALL"FORMAT", device address, volume I.D., 1, 1, owner I.D., master password, 1st, 2nd, 3rd, 4th, and 5th level chains

CAL"FORMAT",2,"LAB",1,1,"GENETICS",
"BLUE",1, 10, 1, 1, 1

Formats a disc and creates a volume label with the following information:

Device address:	2
Volume identification:	LAB
Volume number:	1 (mandatory)
Number in series:	1 (mandatory)
Owner's name:	GENETICS
Master password:	BLUE
1st thru 5th level chains:	1, 10, 1, 1, 1

Also checks for and marks unusable space on the disc. The device must be reserved prior to command execution and released afterwards (CALL"DRES" and CALL"DREL").

CALL"HERRS", device address, retries in last I/O, accumulated retries, successful I/O recoveries, unsuccessful I/O operations

CAL"HERRS",1,A,B,C,D

Requests a count of the last I/O retries, the total I/O retries since last power up, the number of successful recoveries, and the number of unsuccessful operations. These all refer to device 1.

Numbers generated are stored in variables A,B,C and D, respectively.

INIT

Resets variables and closes all files.

INPUT # lfn[,record number]: target variables [,target variables]...

INP #3,40:A$

Reads the entire string (up to a carriage return character) from the beginning of record 40 of lfn 3 and stores it in A$. File must be open and created as random access, ASCII.

INP # 5:B

Reads the first numeric value after the current pointer location in lfn 5 and stores the value in B. The file lfn 5 must be sequential, ASCII, and open.

KILL F.I.[,master password]

KIL"@MYLIBRY # "

Deletes all closed files in MYLIBRY if the library does not have a password.

CALL"MOUNT", device address, target string variable

CAL"MOUNT",1,A$

Activates device 1 for system use and generates a device status message. The message is stored in A$.

CALL"MRKBBG", device address, volume I.D., master password, address of defective space

CAL"MRKBBG",2,"FILE","PASS","010001FA"

This command is executed only after a message specifying defective disc areas appears. The file with the defective area must be deleted before CALL"MRKBBG" is executed. The command specifies a disc in device 2 with a volume ID of FILE and with a master password of PASS. The defective area address is 0001FA (hexadecimal).

CALL"NEXT", lfn, target string variable

CAL"NEXT",3,A$

Closes the current file and opens the next file in a series specified by an OPEN "G" (group) command. As each file in the group is opened, this CALL"NEXT" command assigns the lfn 3 to the file. The command also generates a new file status message with each new file. This message is stored in A$. If LEN(A$)=0, then no files remain in the group.

CAL"NEXT",3," "

Same as previous example except no file status message is generated.

OLD F.I.[,"ASCII"]

OLD"$A/DEPTB"

Locates the binary program file DEPTB in SYSLIB/A and transfers it from the disc to the Graphic System's memory.

OLD F$,"A"

Transfers the ASCII format program in the file identified in F$ from the disc to the Graphic System's memory.

ON EOF (lfn) THEN line number

ON EOF(2)THEN 165

When the end of file is encountered in lfn 2, the system executes a GOSUB 165. Statement RETURN returns control of the program to the line following the READ or INP command that encountered the EOF.

OPEN F.I.[,"G"]; lfn, type of access, target string variable

U = UPDATING (starts at EOF)
R = READ only
F = FULL ACCESS (EOF is marked wherever pointer is at CLOSE.)

OPE"@A/B";1,"U",A$

Opens file B in library A on the current device for updating. This command also associates the file with lfn 1 for use in subsequent commands. The file status message is returned to A$.

OPE"@A # ","G";2,"R",A$

Opens the first file in library A. (Each successive file in library A is opened by CALL"NEXT".) The files are associated with lfn 2 and are designated for read only access. The file status message is returned to A$.

PRINT # lfn [,record number]:data item, data item...

PRI # 3,40:A$

Prints the string denoted by A$ to record 40 in lfn 3. The file must be ASCII, random access, and open.

PRI # 3:"DATA ITEM1 IS 27.2"

Prints the string within quotes after existing data in lfn 3. The file must be open for updating and must be sequential because no record number is specified.

READ # lfn [,record number]:target variables[,target variables]...

REA # 3,40:A$

Reads the binary data in record 40 in lfn 3 and stores it in A$ (random access).

REA # 2:A

Reads the first numeric value after the current pointer location in lfn 2 and stores it in A. File must be open and binary. Because there is no record number in the command, the system requires this to be a sequential file.

CALL"RENAME", device address, old F.I., new F.I.

CAL"RENAME",1,"@DOG/CAT","@DOG/MOUSE"

Renames the file CAT on device 1 to MOUSE.

CAL"RENAME",1,A$,B$

Renames the file named in A$ to the name stored in B$. The file is on device 1.

CALL"REWIND", lfn

CAL"REWIND",1

Re-positions the access pointer for lfn 1 to the beginning of the sequential file.

SAVE F.I.["ASCII"]; [line number [beginning, ending line number]]

SAV"@B/C";100,3150

Transfers the program currently in memory to sequential file C, starting with line 100 and ending with line 3150. The program is saved in binary format. If no file of that name exists, the command automatically creates it.

SAV"@PLOT78","A"

Transfers the entire program currently in memory to the sequential file PLOT78. The program is saved in ASCII form.

SECRET

SEC

Prevents future listing after program has been saved.

NOTE

Use APPEND for ASCII secret files instead of OLD.

CALL"SETTIM", date time

CAL"SETTIM","04-Jul-78 16:30:40"

Sets the system clock to 40 seconds after 4:30 PM on July 4, 1978. The seconds field (:40) is optional.

CALL"SPACE", lfn, requested file size, target numeric variable, target numeric variable

CAL"SPACE",3,3000,A,B

Adjusts the file space in lfn 3 to 3000 bytes. Also generates two numbers: the actual number of bytes required to store the data already in the file and the actual number of bytes allocated. If the data in the file requires 4000 bytes, then the system will not adjust the file size to less than 4000.

The numbers generated by this command are stored in A and B respectively.

CALL"TIME", target string variable

CAL"TIME",A$

Sends the current system time to A$.

TYPE (lfn)

T=TYP(2)

Performs TYPE function on lfn 2. The number returned to variable T identifies one of these six categories:

0	File empty or not open
1	Pointer is at end of file
2	Numeric data or character string data in ASCII format
3	Numeric data in binary format)
4	Character string data in binary format
5	Illegal information

UNIT device address

UNI 3

Specifies device 3 as the current unit.

CAL"UNIT",B

Specifies the device addressed in B as the current device.

CALL"USERLIB", partial F.I.

CAL"USERLIB","PORTLAND/INDUSTRY"

Causes PORTLAND/INDUSTRY to be the current library. Unless this current library is suppressed with special characters (@ or $), the system will attempt to locate all files in PORTLAND/INDUSTRY.

WRITE # lfn[,record number]:data item[,data item]...

WRI #3,40:A$

Writes the binary string in A$ to record 40 in lfn 3 (random file).

WRI #3:A

Writes the numeric value in A to lfn 3 immediately after the current pointer location (sequential file).

NOTES

MANUAL PART NO.
070-2381-01

First Printing JAN 1978
This Printing JAN 1981

REV C, JAN 1981