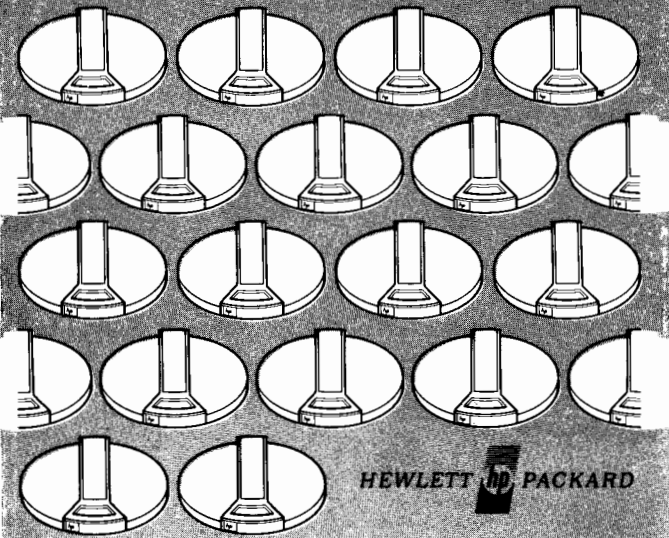


**Quick
Reference
to the
HP 9880
Mass Memory
Subsystem**



HEWLETT  PACKARD

HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

QUICK REFERENCE

The programming information on this card serves as a handy reference for users of the Hewlett-Packard 9830A Calculator - Mass Memory System. Use it as a pocket reference or keep it in a place near your mass memory system.

Turn On

Procedures for turning on your mass memory system are explained in the Mass Memory System Operating Manual. However, here are a few reminders.

- The 11273B Mass Memory ROM must be installed in the highest position of the slots behind the calculator RDM door.
- The 9830A Calculator should be turned ON.
- The 9867 Mass Memory should be turned DN and the DRIVE READY light should be lit.
- Finally, the 11305A Controller should be turned ON.

Warning: If several calculators are connected to a single controller, DO NOT TURN OFF one calculator in the system while another calculator is in use.

Mass Memory Capacity

The Model 9867A Mass Memory contains a single memory cartridge, while the 9867B Mass Memory contains a permanently installed memory platter and an interchangeable cartridge.

	Model 9867A	Model 9867B
Total storage, in bytes	2,433,024	4,866,048
Total storage, in words	1,216,512	2,433,024
Words per record	256	256
Total storage, in records	4,752	9,504
Full-precision numbers per record	64	64
Total storage, in full precision numbers	304,128	608,256
Maximum no. of files	768	1,536

Data Storage Requirements

In order to determine the storage requirements for your data, use the following chart.

Full Precision Numbers	4 words per number
Split Precision Numbers	2 words per number
Integer Precision Numbers	2 words per number
Character Data (strings)	2 characters per word plus 1 additional word per string

Maintenance

The information contained on the mass memory platter is the most valuable item in your system. Your HP service representative can assist you in some cases if the data on your mass memory becomes inaccessible. However, the best insurance for your data and programs is to have a backup copy of this information either in cassette form or as a duplicate platter.

Once your mass memory is put in place, subsequent movements should be minimal. Relocation of your mass memory system should be attempted only with the help of your HP service representative.

There is a strong magnetic field at the rear of the mass memory. Be careful to keep recorded cassettes and other sensitive items away from this area.

Preventive maintenance is required every 60 days in order to ensure the proper performance of your mass memory system.

GENERAL SYSTEM COMMANDS

NAME	EXAMPLES	DESCRIPTION
GET	GET "EDITDR"	Loads the program from the specified file into memory.
	GET BS	
	GET "EDITOR", 50	Renumbers the program lines from the specified file, beginning with the specified line number, and loads the lines into memory.
	GET "EDITOR", 50, 100	Renumbers and loads the program lines and then runs the program, beginning with the specified line number.
GET KEY	GET KEY "BUDGET" GET KEY BS	Loads the definitions of all 20 Special Function keys from the specified file into memory.
CHAIN	CHAIN "PAYROL" CHAIN CS	Loads the program from the specified file into the memory without erasing the current values of variables. A string variable can be used for the file name.
	CHAIN "PAYROL", 40	Renumbers the program lines from the specified file, beginning with the specified line number, and then links the lines into memory.
	CHAIN "PAYROL", 40, 80	Renumbers and links the program lines and then runs the program, beginning with the specified line number.
SAVE	SAVE "SORT" SAVE AS	Stores the entire program on the specified file.
	SAVE "SORT", 100	Stores all lines after and including the specified line.
	SAVE "SORT", 100, 200	Stores lines between and including the specified lines.
SAVE KEY	SAVE KEY "FINAL" SAVE KEY T\$	Stores definitions of Special Function keys on the specified file.
<i>NOTE: The GET, SAVE and CHAIN commands correspond to the LOAD, STORE and LINK commands. String variables can be used for file names in these commands.</i>		
OPEN	OPEN "DATA", 50 OPEN AS, X	Creates a data file of the specified length, in 256-word records. A string variable can be used for the file name, and a variable can be used to specify the file length.
PROTECT	PROTECT "KEY", "*" * PRO "PAY", "OUR"	Assigns the specified protection code to the file so that only those who know the protection code can ASSIGN or KILL the file. The file name and protection code cannot be a string. Also this command is not programmable.
KILL	KILL "DATA"	Erases the specified file.
	KILL "TABLES", "\$\$"	Erases the specified protected file.
	KILL F\$, M\$	A string variable can be used for the file name or protection code.
UNIT	UNIT 0	Selects the specified mass memory unit for subsequent use. The unit number variable must be equal to 0, 1, 2 or 3. If no unit is specified, UNIT 0 is selected. Also, UNIT 0 is selected when a SCRATCH A or LOAD BIN command is executed.
	UNIT X	
<i>NOTE: All effective FILES assignments are erased when the UNIT command is executed.</i>		
CATALOG	CAT CATALOG	Prints information about the size and content of files stored on the mass memory unit. This command is not programmable.

STATEMENTS

NAME	EXAMPLES	DESCRIPTION	NAME	EXAMPLES	DESCRIPTION
FILES	FILES MASTER, UPDATE, *	Reserves space for the specified files. File assignment is completed except for files represented by *. If * appears in the FILES statement, file assignment is completed by means of an ASSIGN statement. Up to 10 files can be named in a FILES statement. The files are referred to by their numeric positions in the FILES list for PRINT # and READ # operations. A new FILES statement obsoletes the previous FILES statement.		READ #1, 2; X, Y READ #F, R; X, Y	Reads the values for the listed variables from the specified record within the file. The record number can be a variable.
	FILES *	Erases file assignments at the end of a program for greater data security.		READ #F, R	Moves the file pointer to the beginning of the specified record.
				READ #F, 1	Resets the file pointer at the beginning of the specified file.
ASSIGN	ASSIGN "DATA", 2, X	Completes file assignment for the specified file. The return variable indicates status of the file: 0 - file is available 3 - file does not exist 4 - file number is out of range	IFEND #	IF END #F THEN 200	Establishes a routine to be followed in case subsequent READ # or PRINT # statements cannot be executed because of 'end of file' or 'end of record' conditions. This statement is programmable only.
	ASSIGN "BUDGET", 3, Y, "SS"	Protected files must be referenced in ASSIGN statements.			NOTE: After an IF END # statement is executed, it remains in effect until a new IF END # statement, FILES or ASSIGN statement, or UNIT command is executed, or until the calculator is turned off.
	ASSIGN FS, I, X, MS	String variables can be used for the file name and protection code. A variable can also be used for the file number.			
		NOTE: After the ASSIGN statement is executed, the file is referenced by its number. However, a FILES statement must exist before the ASSIGN statement is executed.			
		NOTE: File assignment is completed if the return variable for the file is equal to 0. An error occurs if an attempt is made to access an unassigned file.			
PRINT#	PRINT #1; A, B, C PRINT #F; A[1], B, CS	Stores the listed variables sequentially, i.e. serially, in the specified file. The file number can be a variable.	*MAT PRINT #	MAT PRINT #2; A MAT PRINT #3, 2; B	Prints an entire matrix on a file, or on a specified record within a file.
	PRINT #1, 2; X, Y	Stores the listed variables in the specified record of the file. (Record access is known as random access).		MAT PRINT #F, R; A, B, C	Several matrices can be printed in a single operation.
	PRINT #F, R; A, B, C[100]	The record number can be represented by a variable.	*MAT READ #	MAT READ #5; X MAT READ #F, R; M	Reads an entire matrix from a file, or from a specified record within a file.
	PRINT #1; A, B, C, END PRINT #1, 2; X, Y, END	Writes the listed variables followed by a logical 'end of file' mark.		MAT READ #F, R; A, B, C	Several matrices can be read in a single operation.
	PRINT #F, R	Erases the specified record in the file by placing a logical 'end of record' mark at the beginning of the record.		MAT READ #5, J; A(5, 10), B	An implicit REDIM statement can be included in the MAT READ statement.
READ#	READ #1; A(5), B(2, 2), C, DS READ #F; A(5), B(2, 2), C, DS	Reads the values for the listed variables sequentially from the specified file. The file number can be a variable.	FUNCTION		
			TYP	PRINT TYP (1) X = TYP (F) IF TYP(F) < 3 THEN 50 GOTO TYP (F) OF 40, 50, 60 PRINT TYP (-1) X = TYP (-F)	Returns the value to the function as follows: 1. Next item is a full precision number 2. Next item is a character string 3. Next item is 'end of file' 4. Next item is 'end of record' 5. Next item is split precision number 6. Next item is integer precision number
					NOTE: The 'end of record' condition is returned only when the file number is negated; for sequential access, the end of record condition is not significant.
					* The Matrix Operations ROM is required in order to use the MAT PRINT # and MAT READ # statements.

ERROR MESSAGES

MESSAGE	MEANING
ERROR 90	Mass Memory power OFF Controller power OFF Mass Memory drive fault or drive not ready Specified UNIT does not exist Check word or address error Hardware data protect (write not permitted)
ERROR 91	File name or protection code greater than six characters File name or protection code of zero length
ERROR 92	Protected file accessed in FILES statement Incorrect protection code Protection code is not given for protected file Protection code is given for an unprotected file File already protected
ERROR 93	Syntax not valid
ERROR 94	File not found File number reference not valid Record number reference not valid Unit number not valid File not assigned
ERROR 95	Available storage space exceeded Availability table full Directory full
ERROR 96	File size not valid Null program
ERROR 97	File already exists
ERROR 98	Improper file type Improper data type Numeric overflow on data type conversion
ERROR 99	End of file marker reached End of record marker reached

SPECIAL PURPOSE COMMANDS

NAME	EXAMPLES	DESCRIPTION
DCOPY	DCOPY "DAD" TO "SON"	Copies the data contained in the first file into the second file.
	DCOPY "D1",0 TO "D1",1	The files can exist on different units.
		NOTE: The second file must be large enough to accommodate all data contained in the first file. If the first file is protected, the second file must have the same protection code.
DREN	DREN "OLD" TO "NEW"	Renames the first specified file with the second name specified. The contents of the file is unchanged.
	DREN "F1" TO "F2","SS"	For protected files, the protection code must be specified.
DAVTP	DAVTP	Performs a preliminary pack of the data contained in the mass memory availability table. Reorganization of the availability table pointers can sometimes yield space for additional files.
		NOTE: Memory is erased when DAVTP is executed.
DFDUMP	DFDUMP "DATA"	Stores the data records from the specified file onto a cassette. After each 150 records of data are stored, the process stops and the calculator waits for a new cassette to be inserted.
		NOTE: Cassettes containing data written by the DFDUMP command should be carefully labeled for use in the DFLOAD procedure. The cassette must be fully rewound before use, and the cassette should not contain current information since this information will become inaccessible.
DFLOAD	DFLOAD "DATA"	Loads the data records from the cassette on to the specified file. After each 150 records of data are loaded, the process stops and the calculator waits for a new cassette to be inserted.
		NOTE: If the file from which the data was taken is protected, the new file must have the same protection code.



Sales, service, and support in 172 centers in 65 countries.
11000, Colicade 80527 Offices in principal cities throughout the U.S.

NAME	EXAMPLES	DESCRIPTION
DGET	DGET "PROG"	Loads the specified source program into memory and performs BASIC syntax checking. The program is executed immediately.
	DGET "PROG" 0	If the digit 0 follows the file name, the program is not executed. The source program file must not be protected.
DEXP	DEXP I, AS	Converts the value of the specified variable into a 4-digit character string, with leading zeros. (Used to generate line numbers for BASIC statements.)
DBYTE	DBYTE X, AS	Converts the value of the specified variable to its binary equivalent character, stored as a single character in the specified string. (Used to generate quotes for BASIC statements.)