# DATA BASE 2

SHAMMAS SOFTWARE SERVICES 1533F HONEY GROVE DRIVE RICHMOND, VIRGINIA 23229

### DISCLAIMER OF ALL WARRANTY AND LIABILITY

The author of the program makes no warranties, either expressed of implied, with respect to this manual or with respect to the software described in this manual, its quality, performance, merchantability or fitness for any particular purpose. DATA BASE 2 is sold "As Is". While every effort is made to make DATA BASE 2 versatile, in no event will Shammas Software Services and/or the author be liable for direct, indirect, incidental or consequential damages resulting from any defect in the software.

This manual is a copyright of : SHAMMAS SOFTWARE SERVICES 1533F Honey Grove Drive Richmond, Virginia 23229

## TABLE OF CONTENTS

INTRODUCTION	1
ERROR HANDLING	2
USER'S INSTRUCTIONS	4
DB2A	4
DB2B	5
DB2D	7
OPERATING HINTS	9
MEMORY CONFIGURATION	10
COMMAND SUMMARY	11
EXAMPLE	12
EXAMPLE PRINTOUT	14

#### FORWARD

DATA BASE 2 is a format-free data base, for the HP41C, that uses ASCII files in the Extended memory modules to set up the information. It can be regarded as an ASCII file editor and will deal with any ASCII file even if it is generated by other programs. The following hardware is needed or recommended: (1) Hp41C.

(2) The HP-IL interface loop module (#82160A)

(3) The digital cassette drive (#82161A)

(4) The Extended functions module (#82180A)

(5) (recommended) The extended memory module (#82181A).

(6) The HP41C printer (#82143A) or the thermal printer/plotter (#82162A).

All of the hardware products mentioned above are manufactured by the Hewlett-Packard Co.

ACETONE	<	header
-139.0	<	Freezing point (F)
133.16	<	Boiling point (F)
49.3	<	Density (Lb/cuft)

In the above, two sets have been added, the first composed of a header (Benzene) and only two numeric records, while the second record set added is similar to the WATER record set! If this information is what the user wants, DATA BASE 2 will not object to the heterogeneous format of the record sets. DATA BASE 2 will contain the above information in an ASCII file in the following manner:

Rec #	Content
	alaan gilita salip dhun aajaa kalib tilaan qala
0	filename
1	WATER
2	32
3	212
4	62.4
5	BENZENE
6	41.9
7	176.18
8	ACETONE
9	-139.0
10	133.16
11	49.3

Other features of DATA BASE 2 are:

(1) Appending records or characters.

(2) Deleting records or characters.

(3) Inserting records or characters.

(4) Viewing/Printing the files with an option for character count.

(5) Saving/Loading the files to/from mass storage.

DATA BASE 2 consists of three programs. The first two, DB2A and DB2B, are used to create, delete, purge, store, load and edit the ASCII files. The third program, DB2D, has two routine that will recall data from the X-MEM ASCII files (not from mass storage). They can be used as subroutines in a user-program.

An important feature of DATA BASE 2 is that while performing record or character manipulations, the program would like to know where the operation is to be performaed (record pointer position as the Extended-Function manual puts it). This can be one of three options:

(1) The last record dealt with.

(2) A record indicated by its record number (its location in the ASCII file ).

(3) A record indicated by its RELATIVE position in reference to the last header before it.

The last two features are very important. Considering our sample data, the boiling point of water, 212 F, is record number 3 (absolute address or location) and is two records AFTER the header "WATER".

#### ERROR HANDLING AND WARNINGS

As record and character manipulations are performed, DATA BASE 2 monitors the number of characters that are free for additional entries. As records or characters are added there are three possible outcomes, as far as error handling is concerned:

(1) The added data had the adequate place, with at least one free character left. No error messages.

(2) The added data had enough space but left no free characters for more entry. A "FILE FULL" message appears AFTER the operation is performed.

(3) The data exceeded the available space by at least one character. The number of excess characters will be displayed to offer the user a chance to rementer his data after trimming.

The user should be familiar with the Extended-Function module manual supplied with the module. The DATA BASE 2 manual is no substitute for the latter. This manual uses the same terminology found in the module's manual. This terminology is different from that used in the DATA BASE 1 manual and many other data base manuals.

PAGE 4

#### USER'S INSTRUCTIONS

Note 1 : An "R/S#" means to ignore the "R/S" if a printer is used. (1) Load part A of DATA BASE 2. INPUT : [ALPHA] DB2A [ALPHA] FUNCTION : XEQ [ALPHA] READP [ALPHA] ( Optional ) FUNCTION : GTO [.][.] to protect program DB2A from erasure when DB2B is loaded. (1.1) Initialize DB2A. FUNCTION : XEQ [ALPHA] DB2A [ALPHA] (1.2) To create an ASCII file in the X-MEM. FUNCTION : XEQ [ALPHA] A [ALPHA] DISPLAY : FILE SIZE? INPUT : file size (registers) FUNCTION : R/S DISPLAY : FILENAME? INPUT : filename (must be less than 7 characters) FUNCTION : R/S DISPLAY : directory of the X-MEM. (1.3) To purge an ASCII file. FUNCTION : XEQ [ALPHA] [SHIFT] A [ALPHA] ( i.e. XEQ a ) DISPLAY : FILENAME? INPUT : filename. FUNCTION : R/S (1.4) To clear an ASCII file (no purging). FUNCTION : XEQ [ALPHA] [SHIFT] B [ALPHA] (i.e. XEQ b) DISPLAY : FILENAME? INPUT : filename FUNCTION : R/S To save an ASCII (for the first time) file on mass storage (1.5)media. FUNCTION : XEQ [ALPHA] B [ALPHA] DISPLAY : FILENAME? INPUT : filename FUNCTION : R/S (1.6) To overwrite (re-save) an ASCII file with an older version already saved. FUNCTION : XEQ [ALPHA] C [ALPHA] DISPLAY : FILENAME? INPUT : filename FUNCTION : R/S (1.7) To load an ASCII file from mass storage. FUNCTION : XEQ [ALPHA] D [ALPHA] DISPLAY : FILENAME?

INPUT : filename FUNCTION : R/S (2) To load part B of DATA BASE 2. INPUT : [ALPHA] DB2B [ALPHA] FUNCTION : XEQ [ALPHA] READP [ALPHA] (2.1) To initialize and link-up with an ASCII file. This routine can also be used to examine the number of free characters available. FUNCTION : XEQ [ALPHA] DB2B [ALPHA] DISPLAY : FILE? INPUT : filename FUNCTION : R/S DISPLAY : AVLBL CHR=(available chars) For the following operations the user should observe the following notes: Note 2 : In appending, inserting and deleting records or characters the program will prompt for the pointer position as shown below : DISPLAY : POSPT? There are three possible ways to respond : (1) INPUT : none made, indicating the use of the file pointer with its current value. FUNCTION : R/S (2) INPUT : rrr.ccc . This indicates the position by the actual record number (rrr). The (ccc) is used to indicate the character position for character operations. FUNCTION : R/S INPUT : -rrr.ccc . This signals that the position is (3) relative to a header by (rrr) records and (ccc) characters. The program now will ask for the header. DISPLAY : AFTER? INPUT : header name FUNCTION : R/S "NOT FOUND" message is shown if the header name is possibly Α misspelled or nonexistent. Note 3: As indicated earlier, the program will monitor the number of characters available. If the data fits in "on the nose", leaving no more room a "FILE FULL" message is displayed. If the added data exceeds by as much as one character, the excess will be indicated, allowing the user to re-enter the data in a shorter format if that is possible. If not, then it is about time to save the data on mass media. More about this later.

(2.2) To add an alphanumeric header for a record set. FUNCTION : XEQ [ALPHA] A [ALPHA] DISPLAY : HEADER? INPUT : header

FUNCTION : R/S (see note 3) (2.3) To add a numeric record to the set. FUNCTION : XEQ [ALPHA] B [ALPHA] DISPLAY : NUMBER? INPUT : numeric data FUNCTION : R/S (see note 3) (2.4) To append more characters/digits. FUNCTION : XEQ [ALPHA] C [ALPHA] DISPLAY : POSPT? INPUT : pointer (see note 2) FUNCTION : R/S DISPLAY : APPND? INPUT : number to be appended. OR [ALPHA] alphanumeric FUNCTION : R/S (see note 3) (2.5) To delete a record. FUNCTION : XEQ [ALPHA] D [ALPHA] DISPLAY : POSPT? INPUT : pointer (see note 2) FUNCTION : R/S (2.6) To delete characters. FUNCTION : XEQ [ALPHA] [SHIFT] D [ALPHA] (i.e. XEQ d) DISPLAY : POSPT? INPUT : pointer (see note 2) FUNCTION : R/S DISPLAY : NO. CHRS? INPUT : number of characters to be deleted from the current pointer position. FUNCTION : R/S (2.7) To insert a record. FUNCTION : XEQ [ALPHA] E [ALPHA] DISPLAY : POSPT? INPUT : pointer. Insertion is done before the pointer postion (see note 2). FUNCTION : R/S DISPLAY : INSERT REC? INPUT : numeric data OR [ALPHA] alphanumeric data FUNCTION : R/S (see note 3) (2.8) To insert characters. FUNCTION : XEQ [ALPHA] [SHIFT] E [ALPHA] (i.e. XEQ e) DISPLAY : POSPT? INPUT : pointer. Insertion is done before the pointer position (see note 2). FUNCTION : R/S DISPLAY : INSERT CHR? INPUT : numeric data

Page 6

OR [ALPHA] alphanumeric FUNCTION : R/S (see note 3) (2.9) To view/print the ASCII file. FUNCTION : XEQ [ALPHA] [SHIFT] A [ALPHA] (i.e. XEQ a) A loop will start to display the filename and the contents. DISPLAY : record FUNCTION : R/S\* (2.10) To view/print the ASCII file with character count option. FUNCTION : XEQ [ALPHA] [SHIFT] B [ALPHA] (i.e. XEQ b) A loop will start to display the records followed by the number of characters in that record. DISPLAY : record FUNCION : R/S\* DISPLAY : CHRS=(number of characters) FUNCTION : R/S\* At the end of the loop the total number of characters will be shown. DISPLAY : TOTAL=(total)

(3) To use the data base information in a user program.
(3.1) Load the DB2D program.
( Optional ) FUNCTION : GTO [.] [.]
INPUT : [ALPHA] DB2D [ALPHA]
FUNCTION : XEQ [ALPHA] READP [ALPHA]

(3.2) To load one numeric record from the ASCII file, in a user program, into the X-register. The program will use the X, Y and Z registers.

Line Program step -----old file name n PURFL n+1 n+2 new file name n+2 GETAS . . . . . . . . . . . . . . . . header m m+1 rrr.ccc m+2 XEQ DB2C

The first steps above may be needed to purge files no longer needed and load the required ones. Steps m to m+2 will recall a record from the ASCII file into the X-register, by using the relative position (rrr.ccc) after the header.

(3.3) To load a number of numeric records from an ASCII file in X-MEM into a range of registers. The lines m and up in step (3.2) will be:

Page	e 8
------	-----

Line	Program step	
	header	
m+1	bbb.eæe	bbb=1st reg. eee=last reg.
m+2	ENTER^	2
m+3	rrr.ccc	
m+4	XEQ DB2D	

The manual version of applying steps (3.2) and (3.3) is shown in the example.

#### OPERATION HINTS

Having gone through the user's instructions, we present some hints that will be useful:

(1) It may be necessary during data entry to change the display format, either changing the number of digits (say going from FIX 3 to FIX 6) or changing the display mode altogether (say going from FIX 3 to SCI 5). This will affect the number of characters (numerals) stored in the ASCII record. Also clearing flag 29 will eliminate the comma which can occupy one character. This is recommended and will avoid problems if the numeric record is edited by elimiating some numerals and FORGETTING ABOUT THE COMMA.

(2) What do I do when my ASCII file is full? This is a very important question especially when the last entry was not the last record in a record set, thus un-entered data still remains at hand! Two things can be done:

(A) Save the ASCII file on mass storage, purge it and start a new file with another name. The first header should be the last one used in the previous file, followed by the un-entered data. The user should know where his data is by keeping lists of them.

(B) Save the ASCII file on mass storage, purge it and create a larger file with the SAME name. Load the ASCII file from mass storage into the new larger one in X-MEM. Now you can resume entering the rest of the data. This option is not available if there was no extra space in the X-MEM at the beginning. Option A becomes mandatory.

## MEMORY CONFIGURATION

Register #	Content
0	Filename.
1	character counter

Flag #	Use
0	Set : File overflow
1	Set : File barely full
22	Detect numeric entries
23	Detect alpha entries
25	Error detection

LBL DB2A : Initialize DB2 part A. LBL A : Create an ASCII files. LBL B : Save an ASCII file. LBL C : Overwrite an ASCII file. LBL D : Load an ASCII file. LBL a : Purge an ASCII file. LBL b : Clear an ASCII file. LBL DB2B : Initialize and scan ASCII files. LBL A : Input "header" for record sets. LBL B : Input numeric records. LBL C : Append records/characters. LBL D : Delete records. LBL d : Delete characters. LBL E : Insert records. LBL e : Insert characters. LBL DB2C : Load a numeric record to the X-register. LBL DB2D : Load numeric records to a range of registers.

#### EXAMPLE

Returning to our sample data mentioned earlier, we will first enter the data with some deliberate mistakes. Through correcting these mistakes we will show how DATA BASE 2 works. Consider the data to be entred as:

1) Set 1 : WATER , 32 , 212 , 50 . 2) Set 2 : BENZEN , 4211.90 , 176.18 3) Set 3 : ACTONE , -139.00 , 133.16 , 49.30

Let us take the following steps: Set the size to 050. This is extremely generous as far as (1)the needs of DATA BASE 2. Also set the display as FIX 2. The display mode (i.e. what we see) is what is stored. To do away with the comma we will also clear flag 29. This should save one character every time we save a number with an absolute value greater than or equal to one thousand. (2) Load DB2A from the cassette. (3) Initialize the DB2A program. Use (LBL DB2A). (4) Create an ASCII file "CHEM", of size 050. Use (LBL A). (5) Protect the DB2A program by performing a [GT0] [.] [.]. (6) Load program DB2B from the cassette. (7) Initialize and scan file CHEM. Use (LBL DB2B). (8) Enter in the data shown above. For each set. enter the header via (LBL A) and the numeric records that follow it using (LBL B). (9) Display the contents of the file. Use (LBL a). We want to add the 'E' to the word BENZEN. We will (10)append the letter in question to record 5 (the absolute record number of BENZEN). Use (LBL C). (11) Display the updated file contents. Use (LBL a). The third record after the header, WATER, is a wrong number (12)(the density of water is not 50, but 62.4 lb/cuft). We will delete this record using its relative position to the header WATER. Use (LBL C). Display the contents of the updated file. (13) Now let us insert the correct value for the density (i.e. 62.4). This must go before the BENZENE record, now record number 4. Use (LBL E). Display the contents of the updated file. (14) The word ACTONE has an 'E' missing after the second letter. We will correct this by inserting the 'E' character. The absolute record number of ACTONE is 8 and we want to position the pointer at the letter 'T'. The latter has a character position of two (it starts with zero for the 'A'). Thus we will insert the 'E' before the 'T'. Our pointer value will be 8.002, entered in response to the "POSPT" prompt. Use (LBL e). Display the contents of the updated file. (15)The first record after the header BENZENE has a wrong number. The correct number is 41.90 and not 4211.90. One can see that there is an extra '21'. We will use the relative position of this record after BENZENE (one record). We want to delete

starting at character position number 1 and delete two

Thus we will respond to the "POSPT" prompt with a characters. -1.002. Use (LBL d). Display the contents of the updated file. (16) Display the contents of the updated file again, this time with the character count option. Use (LBL b). (17) Protect the program DB2B with a [GTO] [.] [.] . (18) Load program DB2D from the cassette. (19) To be on the safe side, and for good practice, use the FLSIZE command to link with the CHEM ASCII file. Load the third record after the header WATER (20) into the X-register. The value is printed on the listing. (21) Load the second record after the header ACETONE into the X-register and print the result. (22) Load in registers 10 to 12 the numeric records after the header WATER, starting with the first numeric record that follows the header. Print the registers 10 to 12. Here 10.012 is our bbb.eee value.

Page 13

0175 050	49.30 RUN	
SIZE 050	19100 1181	1150.0
FIX 2		XEW U
CE 29		POSPT?
UF 2.2	XFQ a	5 AA DIN
		00000
DB2A		HPPNU?
PEONP		E RUN
		YFO >
XEA "DRSH.	OUTH	nee o
XFO A	UHER	
FILE SIZE?	LOTEP	
50.00 RUN	HT ILS	
FTI ENOME?		CHEM
	32.00	
CHEM KUN	02100	HOTED
CHEM A050		HH I EK
	212.00	
		70 00
GTO	<b>F</b> C 00	02.00
POCKING	36.60	
THORING		212.00
	DENZEN	
DB28	DENLEN	
DEATD		50.00
	4211.90	
XEQ "DB28"	1644470	DENJENE
FILF?		BENZENE
	176.18	
CHER KUN		4011 00
AVLBL CHR=332.00		4211.70
	ACTONE	
		176 18
XEQ A	170.00	110.10
HEADER?	-139.00	
		ACTONE
WHIEK KUN	177 16	
XE@ B	155.10	
NUMBER?		-139.00
1000EX:	49.30	
32.00 KUN		177 1/
XEQ B		133.15
NUMBER2		
		49 70
212.00 KUN		17800
XFQ R		
WIMPEDO		
NURIDEK (		
50.00 RUN		
UFA A		
XEW H		
HEADER?		
RENTEN DIN		
DENZEN KUN		
XFA R		
NUMBER?		
A211 00 DIM		
4211.70 RUN		
XEQ B		
NUMBER?		
NONDER:		
176.18 RUN		
VEO 0		
AEV H		
HEADER?		
ACTONE RIIN		
YFA R		
NUMBER?		
-179 AA PIN		
VEA D		
YEA R		
NUMBER?		
177 16 DIW		
100.10 KUN		
XEQ B		
NUMBER?		

Page 15

POSPT? After? Water	XEQ D -3.00 RUN RUN XEQ a	XEQ E POSPT? 4.00 RUN INSERT REC? 62.40 RUN XEQ a	XEQ e POSPT? 8.002 RUN INSERT CHR? E RUN E RUN XEQ a
CHEM		CHEM	CHEM
NATER		WATER	WATER
32.00		32.00	32.00
212.00		212.00	212.00
BENZENE		62.40	62.40
4211.90		BENZENE	BENZENE
176.18		4211.90	4211.90
ACTONE		176.18	176.18
-139.00		ACTONE	ACETONE
133.16		-139.00	-139.00
49.30		133.16	133.16
		49.30	49.30

POSPT?	-1,001	XEQ d		XEQ b
AFTER? BENZENE NO. CHRS?	1.001	RUN	CHEM Chrs=4.00	CHEM
	2.00	RUN XEQ a	WATER CHRS=5.00	FLSIZE
			32.00 CHRS=5.00	3.00 XEQ "DB2C" 62.40 ***
WATER			212.00 CHRS=6.00	ACETONE 2.00
32.00 212.00			62.40 CHRS=5.00	XEQ "DB2C" 133.16 ***
62.40			BENZENE CHRS=7.00	WATER 10.012 ENTER† 1.00
BENZENE 41.90			41.90 CHRS=5.00	XEQ -DB2D- 10.012 PRREGX
176.18			176.18 CHRS=6.00	R10= 32.00 R11= 212.00 R12= 62.40
HCETUNE -139.00			HCETUNE CHRS=7.00	
133.16 49.30			CHRS=7.00	
			CHRS=6.00	
			CHRS=5.00 Total=68.0	90
				GTO
			PACKING DB2D	READP

DATA BASE 2 is a program package that deals with setting up data records in the ASCII files of the Extended-Memory (X-MEM for short) and saving them om mass storage media, to be retreived later by other programs.

Consider the following data :

WATER	<	header : compund
32	<	Freezing point (F)
212	<	Boiling point (F)
62.4	<	Density (Lb/cuft)

The above is a "record set" starting with an alphanumeric header which indicates that the data following it belongs to water. Three numeric records, some physical pr perties of water, follow the header. This is basically the type of data that DATA BASE 2 is INTENDED to deal with : data that belong to well established (permanent) information, such as chemical data or any other information tabulated in handbooks. DATA BASE 2 is not intended for personal or business data that need one very large file, with information constantly updated. DATA BASE 2 is meant to set up scientific and engineering data that can be elegantly retrieved when other programs are running. DATA BASE 2 has a small routine that can be used for such tasks.

The record set shown earlier is made up of four records. DATA BASE 2 does not impose, at any stage, any format that makes all record sets necessarily contain the same number of records. This freedom is really extended to the fact that the record sets can contain data of different natures : while one record set has physical properties, another record set can contain chemical properties, manufacturing costs...etc. The record sets are seen by DATA BASE 2 as an independent collection of information! This will give the user an absolute control of the data structure and the possibilities of exceeding the scope of the usage description of DATA BASE 2 in this manual. The price is that the user is responsible to set up and structure his data. Should the user desire to use DATA BASE 2 for very specific applications, Shammas Software Services will be glad to be a consultant for him.

Returning to our data mentioned earlier, we can expand to include two more record sets:

WATER	
32	
212	
62.4	
BENZENE	< header
41.9	<pre>&lt; Freezing point (F)</pre>
176.18	<pre>&lt; Boiling point (F)</pre>