

FRED



DEC-12-FZFA-D 1st Printing May 1970

Copyright \bigcirc 1970 by Digital Equipment Corporation

The material in this handbook, including but not limited to instruction times and operating speeds, is for information purposes and is subject to change without notice.

The following are trademarks of Digital Equipment Corporation, Maynard, Massachusetts:

DEC	PDP
FLIP CHIP	FOCAL
DIGITAL	COMPUTER LAB

For additional copies, order DEC-12-FZFA-D from Digital Equipment Corporation, Program Library, Maynard, Mass. 01754. Price \$1.00

TABLE OF CONTENTS

1.0	Progra	am Overview		1
2.0	Envir	onment		1
3.0	Usage			1
4.0	Descr	iption of the Routines	· ·	4
	4.1	GETFLD		4
	4.2	READ		5
	4.3	WRITE		5
	4.4	LOOKUP		6
1	4.5	ENTER		7
	4.6	REPLACE		8
	4.7	FCF		12
	4.8	DELETE		14
5.0	Flowe	harts		15

5.0

1.0 PROGRAM OVERVIEW

FRED (File Replacement, Entry, and Deletion) is a set of PDP-12 subroutines for manipulation of LAP6-DIAL^F indices.

There are four levels of routines, with provision for a routine at any but the lowest level to call any routine of lower level. There is no provision for reentrance or recursion, but the routines are serially reusable (except for REPLACE, as explained later).

Locations $2\emptyset$ to 27 contain DJR, JMP pairs to the entry points of each major routine, so that the coding can be modified without changing calls in external routines.

2.0 ENVIRONMENT

FRED occupies two tape blocks and, when in core, uses four LINC memory blocks (2000₈ words), including space for the index. The routines are segment-independent, but must be loaded at a segment boundary. Thus memory addresses of 2000, 4000, 6000, 14000, etc., can be used, but 2400 or 3000 may not.

In this discussion, all locations are relative to the segment into which FRED is loaded.

3.0 USAGE

3.1 The user's program must load FRED from a DIAL tape, or assemble it with his program, at any memory address which is a multiple of $2\emptyset\emptyset\emptyset_{p}$. It may then be reused until it is overlaid.

3.2 Entry points for the routines of FRED start at location 2Ø of the segment into which FRED is loaded, as follows:

- 2Ø LOOKUP 22 - ENTER 24 - REPLACE 26 - DELETE 3Ø - READ
 - 35 WRITE

¹ LAP6-DIAL-is hereafter referred to as DIAL.

READ and WRITE are called as follows:

LIF X	/SEGMENT INTO WHICH FRED IS LOADED
LDA I	/LOAD AC WITH PARAMETER POINTER
RWPARM	/POINTER TO READ/WRITE PARAMETER LIST
JMP 3Ø (JMP 35)	/DO READ (WRITE)
•	
•	
•	
•	
RWPARM, Y\UNIT	/HIGH-ORDER THREE BITS FOR FIELD
	/LOW-ORDER THREE BITS FOR TAPE UNIT
BUFFER	/12-BIT MEMORY ADDRESS OF DATA
BLOCKNO	/BLOCK NUMBER OF FIRST TAPE BLOCK
COUNT	/NO. OF BLOCKS TO READ/WRITE

The COUNT must not be zero.

Return is to the instruction following the JMP If AC Bit 1 is \emptyset , RWPARM is taken from the caller's instruction segment if 1, the parameter list is in his data segment. Note: There is no check for attempts by the user to write over LAP6-DIAL, nor is there a check to prevent reading over FRED.

3.3	LOOKUP, ENTER, and DELL	ETE are called as follows:
	LIF X	/SEGMENT WITH FRED
	LDA I	/AC: POINTER TO FILE DESCRIPTOR VECTOR
	FDV	/GO TO LOOKUP (ENTER, DELETE)
	JMP 2Ø (JMP 22, 26)	
	•	
	•	
	•	
	•	
	FDV, UNIT	/LINC TAPE Ø-7
	TEXT "NAME????"	/FILE NAME, ENDING WITH 77'S
		/TO FILL FOUR WORDS (8 CHARS)

/ØØ23 FOR SOURCE, ØØØ2 FOR BINARY

TYPE

START

/STARTING BLOCK NO. OF FILE: /FILLED BY LOOKUP, ENTER, REPLACE, OR DELETE

LEN

/LENGTH OF FILE IN BLOCKS: FILLED IN /BY LOOKUP, CALLER MUST SUPPLY IN /ENTER-REPLACE, UNUSED BY DELETE

a) LOOKUP has two returns; the first, immediately following JMP 2 \emptyset , is taken if there is an error in the parameter list, or the named file is not found. The second, two words after JMP 2 \emptyset , is taken if the file is found, indicating that the information in the file descriptor vector is correct.

LIF	X
LDA	I
JMP	LOOKUP
JMP	NOFIND

/GO FIND THE FILE /1ST RETURN FILE DOESN'T EXIST /COME HERE WHEN FILE IS FOUND

b) ENTER has three returns; The first is taken if there already exists a file of the same name and type. The second is taken on errors in parameter list or insufficient space, either in file space or in the index. The third indicates successful updating of the index.

LIF X	/SEGMENT CONTAINING FRED
LDA I	/POINTER TO PARAMETER LIST
FDV	
JMP ENTER	/GO ENTER FILE IN INDEX
JMP EXISTS	/1ST RETURN - FILE ALREADY EXISTS
JMP NOSPACE	/2ND RETURN - NO SPACE FOR FILE
	/COME HERE ON SUCCESSFUL COMPLETION

Note that the largest file which can ever be stored on a DIAL tape is 310 blocks, because that is the length of the largest file area.

c) DELETE has only one return, immediately following the JMP 26.

3.4 REPLACE may be called only immediately after a call to ENTER which took the second return. The parameter list need not be explicitly indicated - REPLACE uses that from the preceding ENTER, But the instruction field must be set again.

There are two returns; the first is taken on error in calling sequence or insufficient space. (This can never occur if the new file is smaller than or equal to the old file). The second indicates successful replacing of the old file entry.

LIF	Х	5	SEGMENT CONTAINING FRED
JMP	REPLAC		/ENTER FOUND A FILE OF SAME NAME
JMP	NOSPAC		/NO SPACE FOR NEW ONE
			/COME HERE ON SUCCESSFUL REPLACE

If REPLACE is not able to find space for a new file, the old file remains intact.

If the call to REPLACE is not immediately preceded by a call to ENTER which returns indicating the file exists, the machine will halt and FRED must be reloaded.

4.0 DESCRIPTION OF THE ROUTINES:

4.1 GETFLD: (Level Ø) called to obtain the address of the the user's parameter list, which he placed in AC before the call. AC is stored in PARAM (Beta 1). The save field buffer is read and stored at SAVFLD. Bit 1 of the parameter list is in the caller's data field, which is also FRED's data field, and GETFLD returns.

If zero, the parameter list is in the caller's instruction field. That field is then obtained from SAVFLD and used to construct an LDF instruction at GTFØlØ, which is executed to set FRED's data field to the caller's instruction field. Bit 1 is set in PARAM and GETFLD exits.

4.2 READ: (Level 1, entry point 3Ø). The return JMP at Ø is saved at RETURN (Beta 17). An RDC instruction is placed in RDWR (Beta 2), and READ jumps to COMMOM. See WRITE.

4.3 WRITE: (Level 1, entry point 35). Return JMP is saved at RETURN (Beta 17), and a WRC instruction is moved to RDWR (Beta 2). From this point (COMMON), READ and WRITE are the same routine. RET2 (Beta 16) is set to 7777 to indicate that the call was from a user outside this field. GETFLD is called to set up PARAM (Beta 1) to point to the parameter list. The next location, RWENT, is the entry point for internal calls by higher lever routines, which must have performed appropriate setup. The 8-mode field (high-order 3 bits of 15-bit address) is obtained from the parameter list and stored at EXT for use in extended addressing. The two high-order bits of the unit number are then moved to AC₁₀₋₁₁, with the low-order bit in Linc. Tape extended operations are then set. The 12 low-order bits of the memory address are obtained from the parameter list and stored at MEMADD. The low-order unit bit is obtained from Link, combined with the READ or WRITE instruction at RDWR, and stored at CMND The first block number is obtained from the parameter list and stored, following the command, at 10BLK.

> The block count, also from the parameter list, is complemented and stored at RDWR (Beta 2) as a loop control.

IOLOOP begins by setting the tape memory address register,

then updating the address at MEMADD. The READ or WRITE instruction is executed, the block number is incremented, and RDWR is incremented to test for completion. If more I/O remains, there is a jump to IOLOOP. Otherwise, RET2 is tested to determine whether the call was internal or external. If internal (RET2 \neq 7777), return is immediated. If external, the user's fields are restored before returning.

4.4 LOOKUP: (Level 2, entry point 20) RET3 (Beta 15) is set to 7777 to indicate external call. Internal calls enter immediately following this point, at LKP $\emptyset \emptyset \emptyset$. Here, the return JMP is saved at RET2 (Beta 16), and GETFLD is called. PARAM (Beta 1) is saved at PARM2 (Beta 3). The current instruction field is obtained, and used to set the address into which the index will be read. Parameters and return address are set up for READ, which is then called at RWENT to bring in the index. Upon return, the index is checked for validity (5757 is first word). If invalid, an empty index is built in core by storing 5757 in each word (from $1\emptyset\emptyset\emptyset$ to 1777), and the error return is taken.

> At LKPØ2Ø, the name in the user's parameter list is compared with each name in the index until a match is found or the end of the index is reached. In the later event, the error return is taken. If a match is found, the type code in the user's parameter list is compared to 23 (S) and 02 (B). If neither, the error return is taken. S causes a jump to WNTS; B jumps to WNTB, which increments the pointer to the index entry by two and flows into WNTS.

Here RET3 is tested for internal or external call; if internal, RET2, the return JMP, is incremented to allow the caller to distinguish between those cases in which there was a successful name match, but no file of the requested type, and those cases in which the name match was unsuccessful. The starting block number of file is then moved to the user's parameter list. The length is then picked up and tested to see whether or not there is a file of the requested type. If not, the length will be 5757, and the error return will be taken. If the requested file exists, RET3 is tested to check for external call. If external, the length is stored. The return address is then incremented to indicate a successful find, and LOOKUP jumps to ERRTN. There, RET3 is tested again. If the call is internal, return is immediate. If external, LOOKUP restores the user's fields before returning.

4.5 ENTER: (Level 3, Entry Point 22) Starting at NTRØØØ, the return JMP is saved in RET3 (Beta 15). LOOKUP is called at its internal entry point LKP $\emptyset \emptyset \emptyset$. Because this is an internal call, there are three returns. The first, indicating that there was no name match, jumps to $NTR\emptyset 2\emptyset$, where MARK (Beta $1\emptyset$) is set to 7777 to indicate no name match, and flows to NTRØ3Ø, which increments the return address, there being no conflict with existing files. From there, control flows into $\text{FSP} \emptyset \emptyset \emptyset$ to find space for the file. Subsequent processing is in common with REPLACE, and is described below. The second return indicates that the name was found, but not with the requested file type. A pointer to the matching entry is saved at MARK, and ENTER jumps to NTRØ3Ø to increment the return JMP. The third return from LOOKUP indicates that the named file exists. A pointer to its index entry is saved at MARK, ENTSW (Beta 11) is set to 1776 to allow a REPLACE to follow, and the first return is taken by going to RTRNØ.

REPLACE: (Level 3, entry point 24) starting at RPLØØØ, the return JMP is saved at RET3 (Beta 15), and ENTSW (Beta 11) is tested for 1776 (indicating that ENTER found a file conflict). Any other value indicates a user error, and the program halts. The LDF instruction at GTFØ1Ø is then moved into the instruction stream to again set the data field appropriately for the user's parameter list, the length field for this file in the index is set to 5757 to eliminate the old file, and REPLACE jumps to FSPØØØ to find space for the file. From this point, processing is in common with ENTER.

The search for file space is performed in two steps; first, a scan is made to find any suitable space in the lower file area. The result, if any, is saved, then a scan of the upper area is made. Because the index is below the middle of the tape, the result of a successful scan of the lower file area can be used to calculate an upper limit for scanning in the upper file area. Beyond this limit, any suitable file space would not be used, since the suitable space in the lower area is closer. Conversely, any space found in the upper area before reaching this limit must be closer to the index than the space found in the lower area. Use of this algorithm eliminates, therefore, the need to compare two possible spaces for closeness to the index, and generally shortens the scan of the upper file area.

Scanning itself is performed by the conflict-search routine, FCT, as follows:

A tentative starting block (TRY) and the length of the desired file (TRYLEN) are set up. Each non-empty index entry is compared to TRY by subtracting its starting block from TRY. If the result is negative, the file specified by this index entry starts at or above TRY. If adding TRYLEN still gives a negative result, the file starts above the end of the tentative file, and there is no conflict. The scan continues to the next index entry.

4.6

If there is a conflict, control is returned to the caller to set a new TRY.

If subtracting the starting block from TRY yields a positive result, TRY is above the file specified by this entry, by the value of the result. In other words, AC contains the distance between TRY and the file concerned. This distance is subtracted from the length of the file in question. Here, a negative result implies no conflict, and the scan continues with the next index entry. A positive result represents a conflict, and control returns to the caller.

During the scan of the lower area, TRY moves downward (away from the index). Each time a conflict is found, a new TRY is calculated by subtracting TRYLEN from the starting block of the file causing the conflict. This is the highest possible starting block which will not cause a conflict with this particular file. The conflict search routine is called again, and the whole process repeated. Thus, the maximum number of iterations is the number of files in the lower file area. During the scan of the upper area, the process is equivalent, except that TRY moves upward. When a conflict is found, a new TRY is calculated as the sum of the start and length of the file in conflict.

In detail, the search is performed as follows:

A SKIP-IF-NEGATIVE instruction (APO I) is moved to FCF \emptyset 5 \emptyset in the conflict search routine to make it ignore any files in the upper file area. A pointer to the start block field of the user's parameter list is saved at LP1 (Beta 8). The user's length request is picked up, tested for validity (zero or negative lengths cause a jump to RTRN \emptyset , indicating error), and saved at TRYLEN. The length is then subtracted from 27 \emptyset , to give the block number of highest starting block in the lower file area which could satisfy the request.

This value is in the AC at FSPØ1Ø, the beginning of the search loop for the lower file area. It is tested to assure that this starting block is positive (i.e., that it is on the tape). If not, there is no space large enough for the file in the lower area, and a jump FSPØ2Ø is taken. If the starting block is positive, it is stored at TRY, and the conflict search routine, FCF, is called. Return is to a JMP FCFØ3Ø if no file in the index would overlap one which started at the block number in TRY. If a conflict is found, the second return is taken from FCF; in this case, TRYLEN is subtracted from the starting block number of the file which conflicts with TRY. This value is in the AC when FSP jumps to FSPØ1Ø to make another search.

The code at FSPØ2Ø is entered, as described above, when TRY goes negative before an acceptable space is found indicating that there is insufficient continuous file space in the lower area for a file the size of TRYLEN. NFSW (Beta 12) is set to zero to indicate this. The last block number on the tape is subtracted from TRYLEN and stored at UPLIM so that, UPLIM contains the complement of the highest possible starting block which would permit a file of the desired size to fit on the tape. FSP then jumps to FSPØ35 to scan the upper file area.

The code at FSPØ3Ø is entered when FCF is unable to find a file which overlaps with one starting at TRY, thus TRY contains the starting block number of a space in the lower file area large enough to accomodate the desired file. NFSW is set to 7777, indicating space was found in the lower area. TRY is saved at SVTRY. The distance between this space and the index is [346 - (TRY + TRYLEN - 1)]. The block as far from the index in the upper file area is this number +347. The complement of the latter result is calculated and stored at UPLIM. Thus the search of the upper file

area can be stopped and considered unsuccessful if no space can be found closer to the index than the space already found in the lower area. At this point control flows into FSPØ35, and processing is the same whether space was found in the lower area or not.

A SKIP-IF-POSITIVE instruction (APO) is moved to FCF \emptyset 5 \emptyset , to cause files in the lower area to be ignored during the conflict search. The AC is initialized to 47 \emptyset , the first block of the upper area, and the upper area scan is begun at FSP \emptyset 4 \emptyset . The AC is stored at TRY, then added to UPLIM. If the result is positive, TRY is too large to be useful because it represents the starting block of a file which would run off the end of the tape, or it is farther from the index than the space found in the lower area. A jump is therefore taken to FSP \emptyset 5 \emptyset , which tests NFSW for a find in the lower area. If none, there is no space, and an error return is taken via RTRN \emptyset . If NFSW is set, however, the starting block at SVTRY is restored to TRY, and control flows to FSP \emptyset 6 \emptyset .

If the sum of TRY and UPLIM was negative or zero (zero result will always be negative), FCF is called to search for a conflict. On finding one, the sum of the starting block and the length of the conflicting file is taken as the next TRY, and FSP jumps to FSPØ4Ø to begin another scan. If no conflict is found, however, before TRY exceeds the absolute value of UPLIM, TRY represents the best starting block for the new file, and FSP jumps to FSPØ6Ø.

At this point, MARK (Beta $l\emptyset$) is tested to determine whether an index entry with the desired name already exists. If so, control transfers to FSPl $\emptyset\emptyset$. If not, the index is scanned for an entry containing 5757 in the name field, indicating it is empty. If no empty entry is found, an error return is taken via RTPN \emptyset .

If an empty entry is found, the file name from the user's parameter list is moved in. The user's type specification (S or B) is examined, and the start and length pointers for the other file type are filled with 5757. Control flows to $FSPl\emptyset\emptyset$, where the starting block and length are stored in the index, and the starting block is stored in the user's parameter list.

The write code, parameter pointer, and return jump are setup for re-writing the index. The I/O handler is called via its internal entry point RWENT. The return address is incremented to indicate successful completion, the user's fields restored, and control is returned.

4.7 FCF: (Level 2, no external entry point) FCF is the conflict search routine. Given a starting block, TRY, and a length, TRYLEN, its task is to scan the index for a file one or more of whose blocks is in the range from TRY to (TRY + TRYLEN - 1). If it should find such a file, pointers to the starting block and length are returned in XPNT and XPNT2, respectively, and control is returned to (P+2), where P is the address of the calling jump. If no conflict is found, control is returned to (P+1).

> Upon entry at FCF $\emptyset\emptyset\emptyset$, the return jump is saved at RET2 and XPNT (Beta 4) is initialized to point to the first index entry. Control flows to FCF \emptyset I \emptyset , where XPNT is incremented and tested for end-of-index. If the end has been reached, there is no conflict and control returns to the caller via RET2. If the end has not been reached, bit 9 of XPNT is tested to determine whether XPNT is pointing to a file name or the start and length area of the entry. In the latter case, control transfers to FCF $\emptyset4\emptyset$. In the former, the name is compared to 5757. If equal, the entry is empty, so XPNT is incremented by 6, and the loop is entered again at FCF $\emptyset1\emptyset$.

If the name is not empty, XPNT is incremented by 4, to address the source-file pointers, and control flows into FCFØ4Ø. FCFØ5Ø having been set to an APO, or APO I instruction, the start block is compared to the index TBLK to determine whether the file is in the wrong area for this scan. If it is the wrong area, the loop is re-entered at FCFØ1Ø. This check, it should be noted, is unnecessary, but was included to speed the scan. Thirteen octal words can be saved by its elimination if space becomes tight.

XPNT2 is set to address the length field, which is tested for validity. If negative, there is no file of that type, and the loop is re-entered at FCFØ1Ø. If the length is positive, the starting block is subtracted from TRY. If the result is negative or zero, TRY is below the start of this file by complement of AC, and control transfers to FCFØ6Ø. If the result is positive, TRY is above the start of this file by the contents of AC. Subtracting this value from the file length gives a positive result if there is a conflict, a negative or zero result if none. If there is a conflict, it is returned to the caller via FCFØ7Ø. If not, the scan is resumed at FCFØ1Ø.

The code at FCFØ6Ø is entered when TRY is below the start of this file. TRYLEN is added to the complement of the block difference. A negative or zero result implies no conflict, and the scan continues at FCFØ1Ø. A positive result is a conflict, so control flows into FCFØ7Ø, which increments RET2 and jumps to it.

(Level 3, Entry Point 26). Beginning at DEL $\emptyset\emptyset\emptyset$, 4.8 DELETE: the return jump is saved at RET3 (Beta 15). LOOKUP is then called via the internal entry point $LKP \emptyset \emptyset \emptyset$ to find the name and file to be deleted. On each of the alternate returns, a JMP RTRNØ is taken, since it is unnecessary to delete a file which doesn't exist. On the third return, the start and length words for this file are filled with 5757 to eliminate the file. Bit $l \emptyset$ of the address of the length word is complemented to give the address of the length word for the other file type; that is, if a source file is being deleted, the low order digit of the address of its length word is 5. Complementing bit 1 gives 7, the address of the binary length. The length of the other type file is tested to determine whether such a file exists. If it is positive, a jump is taken to DELØ1Ø. If negative there is no file of the other type, so the name area of the index entry is set to 5757. At DEL \emptyset 1 \emptyset , the write code, return jump, and parameter pointers are set up, and the I/O handler is called at RWENT to rewrite the index.

5.0 FLOW CHARTS













INDEX

DELETE			1,2,14
ENTER		•	1,2,3,7
Entry Points			1
Environment			1
FCF			12
Flowcharts			15
GETFLD			4
Introduction			1
Loading FRED			1
LOOKUP			1,2,3,6
Memory Addresse	S		1
READ			1,5
REPLACE			1,4,8
			-
Usage			1
WRITE			1,5

900 0		\$ Z Ø	· · · · · · · · · · · · · · · · · · ·	$(\)$
0001		/ FRED FIL	E REPLACEMENT, ENTRY, AND DELETION	\cup
ØØØ2		/ MAR 19, 197	Ø	
0003		1		
0004		1		
0005		/ BETA REGIST	FR DEFINITIONS	
0000	•	/ (13	AND 14 ARE UNUSED)	
0000		/		·
0007				
0010		PARAMEL		
0011	-	RDWR=2		
0012		FDV=2		
0013		PARM2=3		
0014		XPNT=4		
0015		XPNT2=5		
0016		1P1=6		
0017	and the second second	1 02=7		
0017				
0020		PARR-10		
0021		ENISW=11		
0022		NFSW=12		
0023		RET3=15		
0024		RE T2 = 16		
0025		RETURN=17		
0026		1		
0027		/ LOAD ADDRESS OF	DIAL INDEX DO NOT MODIFY: RAMIFICATIONS AR	E OVERWHELMING
aa3a		/		
0034				
0020		1		
0032			DUCTIONS FOR HOF AFTER TOD	
0035		PUP-0 MUDE INST	RULIIUNS FUR USE AFTER IUB	
0034		1		
0035		LRMF = 6244		
0036		LRIB=6234		
0037		LRIF=6224		
0040		1 states and the second		
0041		1 San A		
0042		/ ENTRY POINTS	FOR MAJOR ROUTINES OF FRED	
0043				
0040				
0045	0020 2004			
0045	0020 0000			
0040	0021 0142	JMP LUUKUP		
0047	0022 0006	DJR		
0050	0023 6321	JMP NTRØØØ		
0051	0024 0006	DJR		
0052	0025 6303	JMP RPL000		
0053	0026 0006	DJR		
0054	0027 6641	IMP DEL 000		
0055	000, 0012	FIECT		
		ESECT		
-	ρ	/		
			and the second	

				LAU / WRITE		
0000			/			
0061			/ E	NTERED IN LINC	MOD	DE, WITH AC CONTAINING POINTER TO PARAMETER LIST.
0062			/ I	F BIT 1 IS ZERO	, P	PARAMETERS ARE IN THE CALLERS INSTRUCTION FIELD.
0063	1. 14 A.		/ I	F ONE, PARAMETE	RS	ARE IN HIS DATA FIELD.
0064			/ P	ARAMETER LIST A	SF	OLLOWS
0045			,	ANNELEN EIGT A	•	
0007			, a.		,	
0000			/ 0/	FILLU (3 BI13)	, 	
0007			/ 1/	MEMORY AUDR	£35	(12 B115) /
00/0			/ 2/	FIRST BLUCK	NU	MBER /
0071			/ 3/	NUMBER OF B	LOC	KS /
0072			/			
ØØ73	0030	0057	READ,	SET RETURN	- 7	SAVE RETURN ADDRESS
0074	0031	0000		Ø		
0075	0032	0062		SET I RDWR	1	INDICATE READ OPERATION
0076	0033	0700		700		
0077	0034	6041		JMP COMMON	1	' GO TO COMMON PROCESSING
0100	0035	0057	WRITE.	SET RETURN	1	SAVE RETURN
0101	0036	aaaa		Ø		
0107	0000	0060			,	
0102	0000	0002				INDICATE WITH OF ERATION
0103	0040	0/04	COMMON		,	SET CODE FOR USER CALL
0104	0041	00/0	COMMONI	SEL I REIZ		SET CODE FOR OSER CALL
0105	0042	1177		////		
0106	0043	6116		JMP GETFLD	/	SETUP TO ADDRESS PARAMETERS
0107	0044	1001	RWENT,	LDA PARAM	/	GET FIRST WORD OF PARAMETERS
0110	0045	0306		ROR 6	- 7	' MOVE BANK NO TO RIGHT HALF
Ø111	0046	134Ø		STH	- 7	'HOLD FOR EXTENDED ADDRESSING
Ø112	0047	0054		EXT		
Ø113	ØØ 5Ø	Ø266		ROL I 6	1	' RESTORE UNIT BITS RIGHT
0114	0051	1560		BCL I	1	' KEEP TWO HIGH-ORDER BITS FOR AXO
0115	0052	7774		7774	1	A PLUS LOW-ORDER IN LINK
Ø116	ØØ53	1620		BSF I	1	SET HIGH-ORDER UNIT BITS FOR AXO
0417	0050 0054	0020	EVT	0020		Set fish order only bird for and
6120	0004	0020		470	1	SET EXTENDED ODERATIONS RITS
0120	1077	10001			· ',	SET EXTENDED OFFICATIONS DIAS
0121	0000	1021		LUA I PARAM		GET MEMORT ADDR FROM FARAMS
0122	2007	40/2		SIC MEMADU		HULD FOR TAPE USE
0123	0000	0204		RUL I 4	· '.	GET ONT BIT INTO POSTTION
0124	0061	1600		BSE	/	SET DESIRED READ/WRITE OF CODE
0125	0062	0002		RDWR		
Ø126	0063	4077		STC CMND	/	SET READ/WRITE INSTRUCTION
Ø127	0064	1021		LDA I PARAM	- 7	GET START BLOCK NUMBER
0130	0065	4100		STC IOBLK	- 7	SET APPOROPRIATELY
0131	0066	1021		LDA I PARAM	/	BLOCK COUNT
0132	ØØ67	0017		COM	1	MADE NEGATIVE
Ø133	0070	4002		STC RDWR	1	STORE IN INDEX
0134	0071	1020	IOLOOP.	LDAI	1	GET MEMORY ADDRESS FOR DATA
0135	8872	ดิสิลิลิ	MEMADD.	0	1	ADDRESS FOR NEXT BLOCK
0136	2273	0023	112114007	ΤΜΔ	· /	TELL THE TAPE WHERE
0177	0070	1100			΄,	
013/	0074	01120			<i>'</i> ,	
0140	2100	10700			΄,	CTOPE NEW ADDR
0141	0010	40/2	CHAND	DOC MEMADU		JIUNE NEW AUUN Andreier for recirco operation
0142	1/00	0/00	UMNU	RUL		MUDIFILD FUR DESIRED OPERALIUN
Ø143	Ø1ØØ	0000	TOBEK,	Ø	/	BLUCK NUMBER WANTED
0144	0101	1020		LDA I	/	GET CONSTANT 1
0145	0102	0001		1		
0146	0103	1140		ADM	/	UPDATE BLOCK NUMBER
0147	0104	0100		IOBLK		
0150	0105	0222		XSK I RDWR	1	SKIP IF END OF OPERATION
0151	0106	6071		JMP IOLOOP	1	ELSE CONTINUE
0152		·		EJECT		
-		0	2			
		1.0	,			

P.Z

0154			/ IO IS COMPLET	E RETURN TO CALLER
Ø155			/	
Ø156	0107	Ø216	XSK RET2	/ SKIP IF US CALL
0197	0113	6017	JMP RETURN	/ RETURN NOW 🖵 INTERNAL CALL
0160	0111	0500	IOB	
6161	0112	6244	LRMF	/ RESTORE TO DESIRED STATUS
0162	Ø113	0040	SET Ø	/ RESTORE RETURN JMP
0163	0114	0017	RETURN	
0164	Ø115	6000	JMP Ø	/ GO TO CALLER
Ø165			1	
0166			·/.	
0167			1	
Ø17Ø			/ GETFLD GET C	ALLERS FIELDS AND PARAMETERS
0171			1	
Ø172			/ ENTERED IN LINC	MODE, WITH 10-BIT PARAMETER POINTER IN AC.
Ø173			/ RETURN WITH IND	IRECT POINTER IN LOCATION 1 TO
Ø174			/ DATA FIELD CONT	AINING PARAMETERS.
Ø175		5. 		
Ø176	Ø116	162Ø	GETFLD, BSE I	/ SET BIT Ø (SOON BECOMES BIT 1)
0177	Ø117	4000	4000	
0200	0120	0262	ROL I 2	/ BIT 1 TO LINK
Ø2Ø1	Ø121	0302	ROR 2	/ RESTORE 2-11, SET BIT 1
0202	0122	4001	STC PARAM	/ HOLD AC CONTENTS
0203	Ø123	4011	STC ENTSW	/ CLEAR ENTSW
0204	Ø124	0500	IOB	
0205	Ø125	6234	LRIB	
0206	0126	0303	ROR 3	/ INST FLD TO AC 7-11
0207	Ø127	0452	LZE	/ SKIP IF PARMS IN INST FLD
0210	Ø130	0245	ROL 5	/ DF TO AC 7-11
Ø211	Ø131	1560	BCL I	/ DROP OTHER BITS
Ø212	Ø132	7740	7740	
0213	Ø133	162Ø	BSE I	/ BUILD LDF INST
0214	Ø134	0640	LDF	
0215	0135	4140	STC GTFØ1Ø	/ SAVE IT
Ø216	Ø136	0452	LZE	/ IS DF ALREADY SET?
0217	Ø137	6000	JMP Ø	/ YES
0220	0140	0640	GTFØ1Ø, LDF	/ NO - SET IT
0221	0141	6000	JMP Ø	/ RETURN TO CALLER
0222			FJECT	

P. 3

S

	0224			/ LOOKUP - FIND NAM	ΕD	FILE IN DIAL INDEX	
	0225			1			
	0226			/ ENTERED IN LINC M	ODI	E WITH ADDRESS OF A PARAMETER	
	Ø227			/ LIST IN AC.			
	0230						
	0231			/ 0/ UNIT NUMBER /			
	0232						
	0232			/ I/ TEL NAME /			
	0233			ZZ CUNTU ZZ CONTO			
	0234			7 37 CUNTU /			
	0235			/ 4/ CUNID /			
	Ø236			/ 5/ TYPE (S OR B)/			
	Ø237			/ 6/ STARTING BLOCK			
	Ø24Ø			/ 7/ NO OF BLOCKS /			
	0241			1			
	0242	Ø142	0075	LOOKUP, SET I RET3	1	INDICATE EXTERNAL CALL	
	0243	Ø143	7777	7777			
	Ø244	0144	0056	LKPØØØ, SET RET2	1	SAVE RETURN JMP	
	0245	0145	0000	Ø			
	0246	0146	6116	JMP GETFLD	1	SETUP PARAMETER POINTER	
	0247			/			
	a25a			/ READ THE INDEX			
<i>P</i>	0251			/			
	0252	0147	0043	SET PARM2	1	MOVE PARAMETER POINTER TO ALL AREA	
	0253	0150	0000	DADAM		HOTE FARALLER FOIRFER FO ALT AREA	
	0250	0151	1020		/		
	0254	0151	1020			INT RC	
	0255	0152	0001	LOD			
	0250	0153	0000	108	,	OFT INCTRUCTION FIELD	
	0257	0154	0224		΄,	GET INSTRUCTION FIELD	
	0260	0155	0243	ROL 3	<i>'</i> .	MOVE FIELD NO TO RIGHT HALFWURD	
	Ø2 ⁶¹	Ø156	1340	SIH	/	STORE DATA ADDRESS FOR INDEX READ	
	Ø262	Ø157	0300	XPARM+1			
	Ø263	Ø16Ø	Ø243	ROL 3	1	BANK NO TO BITS 0-2	
	0264	0161	1560	BCL I	/	DROP ALL BUT BANK NO	
	Ø265	0162	Ø777			• •	
	Ø266	Ø163	16ø3	BSE PARM2	/	COMBINE WITH UNIT NO	
	0267	0164	4277	STC XPARM	1	STORE INTO PARAMETER LIST	
	0270	Ø165	0062	SET I RDWR	1	SETUP READ CODE	
	Ø271	Ø166	0700	700			
	0272	Ø167	0077	SET I RETURN	1	SETUP RETURN JMP FROM READ	
	0273	0170	6174	JMP LKPØ10	1	INST MOVED FOR LATER USE	
	0274	0171	0061	SET I PARAM	1	SETUP PARAMETER POINTER	
	9275	0172	0277	XPARM			
	0276	0173	6044	IMP RWENT	1	GO TO READ INDEX	
	0277	0174	0064	IKP010 SET I YENT	,	SO TO READ INDEX	
	0300	0175	1000	TNDEA TNDEA			
	0000 0701	0172	1000		,	CET EIRST WARD OF INDEY	
	0700	0177	1 1 4 4 9		΄,	CUID TE VALID COD INDEV	
	0302	-01//	1400	JAL I	΄,	SAIF IF VALIU FUR INUEA FIDET HODD OF THDEV	
	0303	0200	5/5/	5/5/		FIRST WURD OF INDEX	
	0304	0201	0467	SKP		NUL AN INDEX BUILD UNE	
	0305	0505	6213	JMP LKP020	/	INDEX IS UK GU TU NAME SCAN	
	0306			/			
	0307			/ THERE IS NO INDEX		- CREATE ONE	
	0310			1			
	2311	0203	1020	LDA I	/	GET FILLER WORD	
	2312	0204	5757	5757			
	2313	1205	1244	STA XPNT	1	STORE IN NEXT INDEX WORD	
	2314	0206	0224	XSK I XPNT	1	INCREMENT AND TEST FOR END	
	0315	0207	6205	JMP2	1	ZAP ANOTHER	
	3316	2210	1944	STA XPNT	1	7AP LAST WORD	
	0317	0211	3223	XSK I PARM2		MAKE PARM2 I ALIKE WE SOUGHT MATCH	
	3320	3212	6250		',	RETURN NO EI	
	uu∠u ™ a7.01	- 0616	0222		'		
	0021			EJEUI			

0323			/ SCAN INDEX FOR	NAME	
0324			/		
0325	0213	1023	LKPØ2Ø, LDA I PARM2	/ GET FIRST TRD OF NAME	
0326	0214	4226	STC WORD1	/ SAVE AT COM ARE INST	No. 1997
0327	@215	1020	1 KP030, 1 DA T	CONSTANT 7	
0330	0216	2207	7		
0331	0217	1140	ADM	AND IT TO INDEX POINTER	
0332	0220	Ø004	YPNT	ADD IT TO INDEX TOINTER	
0333	0220	0204	VSK VDNT	A TEST FOR END OF INDEY	
0320	a222	0447		CRID NOT END	
0334	0222	6757		/ SNIP NUL END / STUEPHICE DETHEN NO CHARECC	
0333	0223	1004	JMP ERRIN	/ UTHERWISE RETURN NU SUCCESS	
0000	0224	1024	LUA I XPNI	/ GET A WORD OF INDEX NAME	
0337	0220	1460	SAE I	7 SKIP OUT OF LOOP IF FIRST WORDS EQUAL	
0340	0220	0000	WURDI, Ø		
0341	6221	0215	JMP LKP030	/ LOOP IF NOT EQUAL	
0342	0230	0045	SET XPN12	/ FIRST WORDS EQUAL	
0343	0231	0004	XPNT	/ START FINAL COMPARE	
0344	0232	0042	SET FDV	/ POINT TEMPORARY FDV INDEX	
0345	Ø233	0003	PARM2		
0346	0234	0067	SET I LP2	/ SET LOOP COUNTER	
Ø347	Ø235	7774	- 3		
0350	Ø236	1022	LKP050, LDA I FDV	/ GET NEXT TWO CHARS	
0351	Ø237	1465	SAE I XPNT2	/ COMPARE TO THOSE IN INDEX ENTRY	
Ø352	0240	6215	JMP LKPØ30	/ UNEQUAL - RETURN TO SEARCH LOOP	
0353	Ø241	Ø227	XSK I LP2	/ LOOP TO COMPARE ENTIRE NAME	
0354	0242	6236	JMP LKP050		
0355			. /		
0356			/ IF WE GET HERE	, WEVE FOUND THE NAME	
0357			/		
Ø36Ø	0243	1022	LDA I FDV	/ PICK UP SOURCE/BINARY CODE	
0361	0244	1420	SHD I	/ IS THE CODE S?	
0362	0245	2300	2300		
0363	0246	6263	JMP WNTS	/ YES - GIVE HIM SOURCE	
0364	0247	1420	SHD I	/ IS THE CODE B?	
0365	0250	0200	0200	In the Sour D.	
Ø366	0251	62.61	IMP WNTR	/ YES - CIVE HIM BINARY	
0367					
0370			/ COME HERE ON D	ETECTING ERROR. OR UNSUCCESSEUL FIND	
0371			/		
0372	0252	Ø215	FRRTN. XSK RET3	/ TEST FOR INTERNAL CALL	
0373	0253	6016	IMP RET2	/ RETURN NOW TE INTERNAL	
0374	0254	0040	SET Ø	/ MOVE IN RETURN IMP	
0375	0255	0016	RET2		
0376	0256	0500	TOB		
0377	0257	6244	I RMF		
9499	0260	6000	IMP Ø	RETURN TO CALLER	
0401	2202	0000	EIECT	VALIONA TO OALLEN	
-			LOLOT		
		0	5		
		· / ·			· · ·
					·

0403				/ W	EVE FOUND WHAT H	ΕI	WANTS - GIVE IT HIM
0404			0005	/	YOK I YONTO	,	CANT OWED DUT THE
0405		0261	0225	WNIB,	XSK I XPNIZ	1	CANT SKIP; BUT THE DOLUTED
0406		0262	0225	_	XSK I XPNT2	/	FAKES UUT THE PUINTER
0407		Ø263	Ø215	WNTS,	XSK RET3	/	IF INTERNAL CALL
0410		0264	0236	÷	XSK I RET2	/	SETUP FOR THREE RETURNS
0411		Ø265	1025		LDA I XPNT2	1	GET STARTING BLOCK NO FROM INDEX
0412		Ø266	1062		STA I FDV	/	STORE IN CALLERS PARAMETER LIST
Ø413		Ø267	1025		LDA I XPNT2	/	GET NO OF BLOCKS
0414		0270	Ø451		APO	1	SKIP IF DESIRED DATA EXISTS
0415		Ø271	6252		JMP ERRTN	1	RETURN ERROR IF NO FILE
0416		Ø272	Ø215		XSK RET3	/	DONT STORE RESULT IF INTERNAL
0417		Ø273	0467		SKP		
0420		0274	1062		STA I FDV	1	RETURN NO OF BLOCKS
0421		Ø275	Ø236		XSK I RET2	1	INCREMENT JMP RETURN ADDRESS
Ø422		Ø276	6252		JMP ERRTN	1	NOT REALLY AN ERROR
0423		~ 2 · -		1	•		
0424				/ P	ARAMETERS TO REAL	DF	ROUTINE
Ø425				1			
Ø426		Ø277	0000	XPARM.	Ø	1	FIELD AND UNIT
0427		0300	1000		INDEX	1	DATA ADDRESS
0430	•	0301	0346		346	1	BLOCK NUMBER
0431		0302	0002		2	1	BLOCK COUNT
0432			~~~~		FJECT		
-							
		سکم .	16				
				0	6		
				P	, 0		
				r			

1 REPLACE -- REPLACE A NAMED ENTRY IN DIAL INDEX 0434 1 0435 1 ENTERED IN LINC MODE AFTER CALONG ENTER. MAY BE CALLED ONLY IMMEDIATEL OFTER ENTER HAS RETURNED TO P+1, 0436 Ø 0437 INDICATING THAT A FILE OF THE PROPOSED NAME AND TYPE ALREADY EXISTS. 0440 0441 0442 0303 0055 RPL000, SET RET3 / SAVE RETURN 0443 0304 0000 Ø 0444 0305 Ø231 XSK I ENTSW / TEST FOR NAME FOUND, SET "REPLACE" 0445 Ø3Ø6 RPLØ10, HLT / ILLEGAL SEQUENCE -- STOP 0000 0307 0446 Ø211 XSK ENTSW / TEST FOR CONTINUE AFTER HALT 0310 6306 / BAD BOY -- STOP AGAIN 0447 JMP RPLØ10 Ø311 / PICK UP LDF INSTRUCTION Ø45Ø 1000 LDA Ø451 Ø312 Ø14Ø GTFØ10 Ø452 Ø313 4314 STC .+1 Ø453 2314 0000 Ø / BECOMES AN LDF INSTRUCTION 8315 1020 0454 LDA I / ZAP LENGTH FIELD OF THIS FILE Ø455 2316 5757 5757 \$317 1045 Ø456 STA XPNT2 0457 1328 6345 JMP FSP000 / ALL LOOKS GOOD -- DO YOUR THING 0460 Ø461 ENTER -- ADD A NAMED ENTRY TO DIAL INDEX 1 Ø462 1 ENTERED IN LINC MODE WITH AC POINTING TO A PARAMETER LIST 0463 Ø464 IDENTICAL TO THAT FOR LOOKUP, EXCEPT THAT THE FILE-LENGTH 0465 FIELD IS FILLED BY THE USER. Ø466 Ø467 RETURN IMMEDIATELY FOLLOWING THE JMP (P+1) IF THE FILE ALREADY EXISTS. 0470 RETURN TO P+2 IF THERE IS NO ROOM FOR THE FILE. 0471 RETURN TO P+3 IF OPERATION COMPLETE (DIAL INDEX HAS BEEN UPDATED). Ø472 0473 0321 0055 NTRØØØ, SET RET3 / SAVE RETURN 0474 0322 0000 Ø 0475 0323 6144 JMP LKPØØØ / LOOKUP NAME IN INDEX Ø476 0324 6342 JMP NTRØ20 / THIS NAME IS NOT IN INDEX Ø477 0325 6335 JMP NTRØ1Ø / NAME IS IN INDEX, BUT NOT WITH THIS TYPE 0500 0501 1 WE FOUND A FILE OF THIS NAME AND TYPE 0502 1 0326 1020 0503 LDA I / GET CONSTANT -2 0327 7775 0504 -2 ADD XPNT2 0505 0330 2005 / POINT TO POINTER AREA - 2 0506 Ø331 4010 STC MARK / SAVE POINTER FOR REPLACE 0507 0332 0071 SET I ENTSW / INDICATE READY-FOR-REPLACE Ø51Ø 0333 1776 1776 0511 0334 6543 JMP RTRNØ / GO HOME Ø512 1 0513 1 FOUND THE NAME, BUT NOT TYPE Ø514 0335 1020 0515 NTRØ10, LDA I / CONSTANT -2 0516 0336 7775 -2 Ø517 0337 2005 ADD XPNT2 / FROM POINTER INDEX 0520 0340 4010 STC MARK / SAVE ADDR OF FILE POINTERS 0521 0341 6344 JMP NTRØ30 / GO BEGIN SCAN FOR FILE SPACE Ø522 1 0523 1 NOTHING FOUND IN THE LOOKUP SCAN 0524 0070 Ø525 0342 NTRØ2Ø, SET I MARK / INDICATE NO FIND 0526 0343 7777 7777 0344 0235 NTRØ30, XSK I RET3 Ø527 / NO NAME CONFLICT, INCREMENT RETURN 0530 EJECT

2 7

	Ø532			/ F	SP - FIND SP	ACE ON	DIAL TAPE FOR NEW FILE
	Ø533			1			
	0534	Ø345	1020	FSPØØØ,	LDA I	1	PICK UP SKIP-NEG INSTRUCTION
	0535	0346	0471		APO I	1	MOVED TO ECE050
	0536	0347	4605		STC FCF050	1	STORE IN INDEX-SCAN ROUTINE
	0537	0350	1020		IDA T		
	0540	0351	0005		5		
	0541	0352	2003		ADD PARMS	/	POINT TO USERS START BLOCK ETELD
	0540	0353	1040		STA	<i>'</i> ,	HOLD EOR LATER
	0542	0350	0004				HULD FOR LATER
	0543	0324	0000			,	CTODE BOINTED
	0544	0399	1007		SIG LPZ		
	0242	0320	1021		LUA I LPZ.		FICK UP LENGTH
	0540	0351	0451		APU DIDNA		IEST FUR PUSITIVE LEN REQUEST
	054/	0300	0243		JMP RIRNØ	. /	ELSE REFURN ERRUR
	0550	0301	04/0		AZE I		SKIP IF LEN NUT ZERU
	0551	0362	6543		JMP. RIRNØ.		BOMB IF NULL REQUEST
	0552	0363	1040		SIA	/	HOLD LENGTH HANDY
	Ø553	Ø364	0636		TRYLEN		
	Ø554	Ø365	0017		COM		
	Ø555	Ø366	1120	· .	ADA I	/	SUBTRACT LEN FROM HIGHEST BLOCK NO, LOW FILE
	0556	Ø367	0270		27Ø		
	Ø557	Ø37Ø	Ø47Ø	FSP010,	AZEI	/	SKIP NOT ZERO
	Ø56Ø	Ø371	0011		CLR	/	FORCE TRUE ZERO IF RESULT IS 7777
	Ø561	Ø372	Ø451		APO	1	SKIP IF STILL ON THE TAPE
	0562	Ø373	64Ø4		JMP FSPØ2Ø	1	NO SPACE IN LOW FILE AREA
	Ø563	Ø374	4635		STC TRY	/	HOLD TRIAL STARTING BLOCK
	0564	Ø375	655Ø		JMP FCFØØØ	1	GO FIND POSSIBLE CONFLICT
	Ø565	0376	6413		JMP FSP030	1	HOORAY NO CONFLICT
	0566	0377	1000		IDA	1	NOPE THAT TRY IS NO GOOD
	0567	0400	Ø636		TRYLEN		
	0570	0401	0017		COM	/	SUBTRACT THE SEARCH LENGTH FROM
	0571	0402	1104		ADA XPNT		THE START BLOCK OF CONFLICT FILE
	Ø572	0403	6370		INP ESPOIN	,	AND TRY AGAIN
	0573	0100	0070	/	SHI I SI DID		
	0574			/ u	EVE FOUND NO	SPACE	LARCE ENGLICH IN THE LOWER ETLE AREA
	0575			/		JINCL	LANGE ENDOON IN THE EONEN TILE ANEA
	0576	0404	ØØ72	ESPASA.	SET T NESH	,	SET NO_EIND SWITCH
	0577	0404	00,2	1310201	0		SET NOTTIND SHITCH
	0577	0405	10100			,	DICK HE MINHS END OF TAPE
	0601	0400	6777			,	FICK OF MINOS END OF TAPE
	0001	0107	2676		ADD TRYLEN	,	NTALLE LAST FEASTER STADT DLOCK
	0002	0410	2000		ADD INILEN	· · ·	HOLD AS SCAN LIMIT
	0003	0411	4040		SHU UFLIM		HULU AS SCAN LIMIT
	0004	0412	0425	,	JMP FSP035		ENTER UPPER SUAN
	0000			/ T	NY DOINTO TO	0000	
	0000				RT PUINIS IU	6000 3	SPACE IN LUWER AREA
	0007	a 4 4 7	a a70	500070	OFT THERN	,	
	0010	0413	2100	F SPUSU,	SET I NESW	/	INDICATE SPACE FOUND
	0011	0414	1///		////	. ,	
	0612	0415	1000		LDA	/	PICK UP ULD TRY
	0013	0410	0035		IRT	·	
	0014	0417	1040		SIA	. /	SAVE II
	0015	0420	003/		SVIRY		
	Ø616	0421	2636		ADD TRYLEN		GET END BLOCK
	0617	0422	1120		ADA I	/	GET COMPARE CONSTANT (TRYEND-2*INDEX LOC)
	0620	0423	7061		-716		
	Ø621	0424	4640		STC UPLIM	/	MINUS HIGHEST DESIRABLE STARTING BLOCK
	0622	0425	1020	FSP035,	LDA I	/	GET SKIP-POS INSTRUCTION
**	0623	0426	Ø451		APO	1	MOVED TO FCF050
	0624 👝	0427	4605		STC FCFØ5Ø	/	STORE IN INDEX SCAN ROUTINE
	Ø625 🦳	0430	1020		LDA I	1	INITIAL ST MING BLOCK, UPPER FILE
	0626 👐	0431	0470		470		
	Ø627	0432	1040	FSP040,	STA	/	SET NEW TRY BLOCK
	-	1 - AN		÷ .	£8		

$\begin{array}{cccccccccccccccccccccccccccccccccccc$	APO I JMP FSP050 JMP FCF000 JMP FSP060 LDA XPNT ADA XPNT2 JMP FSP040 / / THERE WAS NO SPA / FSP050, XSK NFSW JMP RTRN0 LDA SVTRY STC TRY / / THERE IS SPACE A / FSP060, XSK MARK JMP FSP100 / / WE MUST SEARCH F / SET I MARK INDEX FSP070, LDA I 7 ADM MARK XSK MARK	<pre>/ SKIP IF STILL IN USEFUL REGION / NO SPACE</pre>
43.6 64.44 43.7 65.50 44.0 64.51 44.1 1.0.0.4 44.2 1.1.0.5 44.3 64.32 44.4 0.2.1.2 44.5 6.5.4.3 44.6 1.0.0.0 44.7 0.6.3.5 45.1 0.2.1.0 45.2 6.5.2.5 45.3 0.0.7.0 45.4 1.0.0.0 45.5 1.0.0.0 45.5 1.0.0.0 45.5 1.0.0.0 45.5 1.0.0.0 45.5 1.0.0.0 45.6 0.0.7.0 45.7 1.1.4.0.0 45.0 0.0.1.0	JMP FSP050 JMP FCF000 JMP FCF000 JMP FSP060 LDA XPNT ADA XPNT2 JMP FSP040 / / / THERE WAS NO SPA / / / THERE WAS NO SPA / / / THERE WAS NO SPA / / / THERE IS SPACE A / / / THERE IS SPACE A / / / THERE IS SPACE A / / / / THERE IS SPACE A / / / / THERE IS SPACE A / / / / THERE IS SPACE A / / / / / / / / / / / / / / / / / / /	<pre>/ NO SPACE ONFLICT / SEARCH FOR ONFLICT / WEVE FOUND SPACE / THIS TRY WONT WORK /SO TRY AT END OF CONFLICT FILE ACE FOUND IN UPPER FILE AREA / SKIP IF FOUND IN LOWER AREA / RETURN BAD NEWS NO SPACE / GET START BLOCK OF FILE IN LOWER ARE. / THATS THE ONE WELL USE AT TRY / WAS THERE A FILE BY THIS NAME / YES - GO USE THAT ENTRY FOR EMPTY INDEX SPACE / INITIALIZE MARK POINTER / BUMP INDEX INDEX</pre>
437 6550 440 6451 441 1004 442 1105 443 6432 444 0212 445 6543 446 1000 447 0637 450 4635 451 0210 452 6525 453 0070 454 1000 455 1020 456 0010 461 0210	JMP FCFØØØ JMP FSPØ6Ø LDA XPNT ADA XPNT2 JMP FSPØ4Ø / / THERE WAS NO SPA / FSPØ5Ø, XSK NFSW JMP RTRNØ LDA SVTRY STC TRY / / THERE IS SPACE A / FSPØ6Ø, XSK MARK JMP FSP1ØØ / / WE MUST SEARCH F / SET I MARK INDEX FSPØ7Ø, LDA I 7 ADM MARK XSK MARK	<pre>/ SEARCH FOR SONFLICT / WEVE FOUND SPACE / THIS TRY WONT WORK /SO TRY AT END OF CONFLICT FILE ACE FOUND IN UPPER FILE AREA / SKIP IF FOUND IN LOWER AREA / RETURN BAD NEWS NO SPACE / GET START BLOCK OF FILE IN LOWER ARE. / THATS THE ONE WELL USE AT TRY / WAS THERE A FILE BY THIS NAME / YES - GO USE THAT ENTRY FOR EMPTY INDEX SPACE / INITIALIZE MARK POINTER / BUMP INDEX INDEX</pre>
440 6451 441 1004 442 1105 443 6432 444 0212 445 6543 446 1000 447 0637 450 4635 451 0210 452 6525 453 0070 454 1000 455 1020 456 0070 457 11400 456 0010 461 0210	JMP FSP060 LDA XPNT ADA XPNT2 JMP FSP040 / / THERE WAS NO SPA / FSP050, XSK NFSW JMP RTRN0 LDA SVTRY STC TRY / / THERE IS SPACE A / FSP060, XSK MARK JMP FSP100 / WE MUST SEARCH F / SET I MARK INDEX FSP070, LDA I 7 ADM MARK XSK MARK	<pre>/ WEVE FOUND SPACE / THIS TRY WONT WORK /SO TRY AT END OF CONFLICT FILE ACE FOUND IN UPPER FILE AREA / SKIP IF FOUND IN LOWER AREA / RETURN BAD NEWS NO SPACE / GET START BLOCK OF FILE IN LOWER ARE / THATS THE ONE WELL USE AT TRY / WAS THERE A FILE BY THIS NAME / YES - GO USE THAT ENTRY FOR EMPTY INDEX SPACE / INITIALIZE MARK POINTER / BUMP INDEX INDEX</pre>
441 1004 442 1105 443 6432 444 0212 445 6543 446 1000 447 0637 450 4635 451 0210 452 6525 453 0070 454 1000 455 1020 456 007 457 1140 461 0210	LDA XPNT ADA XPNT2 JMP FSP040 / / THERE WAS NO SPA / FSP050, XSK NFSW JMP RTRN0 LDA SVTRY STC TRY / / THERE IS SPACE A / FSP060, XSK MARK JMP FSP100 / WE MUST SEARCH F / SET I MARK INDEX FSP070, LDA I 7 ADM MARK XSK MARK	<pre>/ THIS TRY WONT WORK /SO TRY AT END OF CONFLICT FILE ACE FOUND IN UPPER FILE AREA / SKIP IF FOUND IN LOWER AREA / RETURN BAD NEWS NO SPACE / GET START BLOCK OF FILE IN LOWER ARE / THATS THE ONE WELL USE AT TRY / WAS THERE A FILE BY THIS NAME / YES - GO USE THAT ENTRY FOR EMPTY INDEX SPACE / INITIALIZE MARK POINTER / BUMP INDEX INDEX</pre>
442 1105 443 6432 444 0212 445 6543 446 1000 447 0637 450 4635 451 0210 452 6525 453 0070 454 1000 455 1020 455 1020 455 1020 456 0007 457 1140 461 0210	ADA XPNT2 JMP FSP040 / / THERE WAS NO SPA / FSP050, XSK NFSW JMP RTRN0 LDA SVTRY STC TRY / / THERE IS SPACE A / FSP060, XSK MARK JMP FSP100 / / WE MUST SEARCH F / SET I MARK INDEX FSP070, LDA I 7 ADM MARK XSK MARK	<pre>/SO TRY AT END OF CONFLICT FILE ACE FOUND IN UPPER FILE AREA / SKIP IF FOUND IN LOWER AREA / RETURN BAD NEWS NO SPACE / GET START BLOCK OF FILE IN LOWER ARE / THATS THE ONE WELL USE AT TRY / WAS THERE A FILE BY THIS NAME / YES - GO USE THAT ENTRY FOR EMPTY INDEX SPACE / INITIALIZE MARK POINTER / BUMP INDEX INDEX</pre>
 443 6432 444 0212 445 6543 446 1000 447 0637 450 4635 451 0210 452 6525 453 0070 454 1000 455 1020 456 0007 456 0010 460 0010 9210 	JMP FSP040 / THERE WAS NO SPA FSP050, XSK NFSW JMP RTRN0 LDA SVTRY STC TRY / THERE IS SPACE A / FSP060, XSK MARK JMP FSP100 / WE MUST SEARCH F SET I MARK INDEX FSP070, LDA I 7 ADM MARK XSK MARK	ACE FOUND IN UPPER FILE AREA / SKIP IF FOUND IN LOWER AREA / RETURN BAD NEWS NO SPACE / GET START BLOCK OF FILE IN LOWER ARE / THATS THE ONE WELL USE AT TRY / WAS THERE A FILE BY THIS NAME / YES - GO USE THAT ENTRY FOR EMPTY INDEX SPACE / INITIALIZE MARK POINTER / BUMP INDEX INDEX
444 0212 445 6543 446 1000 447 0637 450 4635 451 0210 452 6525 453 0070 455 1020 455 1020 455 1020 456 0010 460 0010 461 0210	/ THERE WAS NO SPA / FSP050, XSK NFSW JMP RTRN0 LDA SVTRY STC TRY / / THERE IS SPACE A / FSP060, XSK MARK JMP FSP100 / / WE MUST SEARCH F / SET I MARK INDEX FSP070, LDA I 7 ADM MARK XSK MARK	ACE FOUND IN UPPER FILE AREA / SKIP IF FOUND IN LOWER AREA / RETURN BAD NEWS NO SPACE / GET START BLOCK OF FILE IN LOWER ARE / THATS THE ONE WELL USE AT TRY / WAS THERE A FILE BY THIS NAME / YES - GO USE THAT ENTRY FOR EMPTY INDEX SPACE / INITIALIZE MARK POINTER / BUMP INDEX INDEX
444 0212 445 6543 446 1000 447 0637 450 4635 451 0210 452 6525 453 0070 454 1000 455 1020 456 0007 457 1140 460 0010 461 0210	<pre>/ THERE WAS NO SPA / FSP050, XSK NFSW JMP RTRN0 LDA SVTRY STC TRY / / THERE IS SPACE A / FSP060, XSK MARK JMP FSP100 / / WE MUST SEARCH F / SET I MARK INDEX FSP070, LDA I 7 ADM MARK XSK MARK</pre>	ACE FOUND IN UPPER FILE AREA / SKIP IF FOUND IN LOWER AREA / RETURN BAD NEWS NO SPACE / GET START BLOCK OF FILE IN LOWER ARE / THATS THE ONE WELL USE AT TRY / WAS THERE A FILE BY THIS NAME / YES - GO USE THAT ENTRY FOR EMPTY INDEX SPACE / INITIALIZE MARK POINTER / BUMP INDEX INDEX
444 0212 445 6543 446 1000 447 0637 450 4635 451 0210 452 6525 453 0070 454 1000 455 1020 456 0010 460 0010 461 0210	/ FSP050, XSK NFSW JMP RTRN0 LDA SVTRY STC TRY / / THERE IS SPACE A / FSP060, XSK MARK JMP FSP100 / / WE MUST SEARCH F / SET I MARK INDEX FSP070, LDA I 7 ADM MARK XSK MARK	<pre>/ SKIP IF FOUND IN LOWER AREA / RETURN BAD NEWS NO SPACE / GET START BLOCK OF FILE IN LOWER ARE / THATS THE ONE WELL USE AT TRY / WAS THERE A FILE BY THIS NAME / YES - GO USE THAT ENTRY FOR EMPTY INDEX SPACE / INITIALIZE MARK POINTER / BUMP INDEX INDEX</pre>
444 0212 445 6543 446 1000 447 0637 450 4635 451 0210 452 6525 453 0070 455 1020 455 1020 455 1020 455 1020 455 0010 456 0010 460 0210	FSP050, XSK NFSW JMP RTRN0 LDA SVTRY STC TRY / / THERE IS SPACE A / FSP060, XSK MARK JMP FSP100 / / WE MUST SEARCH F / SET I MARK INDEX FSP070, LDA I 7 ADM MARK XSK MARK	<pre>/ SKIP IF FOUND IN LOWER AREA / RETURN BAD NEWS NO SPACE / GET START BLOCK OF FILE IN LOWER ARE / THATS THE ONE WELL USE AT TRY / WAS THERE A FILE BY THIS NAME / YES - GO USE THAT ENTRY FOR EMPTY INDEX SPACE / INITIALIZE MARK POINTER / BUMP INDEX INDEX</pre>
445 6543 446 1000 447 0637 450 4635 451 0210 452 6525 453 0070 454 1000 455 1020 456 0007 457 1140 460 0010 461 0210	JMP RTRNØ LDA SVTRY STC TRY / THERE IS SPACE A / FSPØ6Ø, XSK MARK JMP FSP1ØØ / / WE MUST SEARCH F / SET I MARK INDEX FSPØ7Ø, LDA I 7 ADM MARK XSK MARK	<pre>/ RETURN BAD NEWS NO SPACE / GET START BLOCK OF FILE IN LOWER ARE / THATS THE ONE WELL USE AT TRY / WAS THERE A FILE BY THIS NAME / YES - GO USE THAT ENTRY FOR EMPTY INDEX SPACE / INITIALIZE MARK POINTER / BUMP INDEX INDEX</pre>
446 1000 447 0637 450 4635 451 0210 452 6525 453 0070 454 1000 455 1020 455 0007 457 1140 460 0010 461 0210	LDA SVTRY STC TRY / THERE IS SPACE A / FSPØ60, XSK MARK JMP FSP1ØØ / / WE MUST SEARCH F / SET I MARK INDEX FSPØ70, LDA I 7 ADM MARK XSK MARK	<pre>/ GET START BLOCK OF FILE IN LOWER ARE / THATS THE ONE WELL USE AT TRY / WAS THERE A FILE BY THIS NAME / YES - GO USE THAT ENTRY FOR EMPTY INDEX SPACE / INITIALIZE MARK POINTER / BUMP INDEX INDEX</pre>
 447 Ø637 450 4635 451 Ø21Ø 452 6525 453 ØØ7Ø 454 1ØØØ 455 1Ø2Ø 455 ØØØ7 457 114Ø 460 ØØ1Ø 461 Ø21Ø 	SVTRY STC TRY / THERE IS SPACE A / FSPØ6Ø, XSK MARK JMP FSP1ØØ / / WE MUST SEARCH F / SET I MARK INDEX FSPØ7Ø, LDA I 7 ADM MARK XSK MARK	 / THATS THE ONE WELL USE AT TRY / WAS THERE A FILE BY THIS NAME / YES - GO USE THAT ENTRY FOR EMPTY INDEX SPACE / INITIALIZE MARK POINTER / BUMP INDEX INDEX
 450 4635 451 0210 452 6525 453 0070 454 1000 455 1020 456 0010 461 0210 	STC TRY / THERE IS SPACE A / FSPØ6Ø, XSK MARK JMP FSP1ØØ / WE MUST SEARCH F / SET I MARK INDEX FSPØ7Ø, LDA I 7 ADM MARK XSK MARK	<pre>/ THATS THE ONE WELL USE AT TRY / WAS THERE A FILE BY THIS NAME / YES - GO USE THAT ENTRY FOR EMPTY INDEX SPACE / INITIALIZE MARK POINTER / BUMP INDEX INDEX</pre>
451 0210 452 6525 453 0070 454 1000 455 1020 456 00010 460 0010 461 0210	/ THERE IS SPACE A / FSPØ6Ø, XSK MARK JMP FSP1ØØ / / WE MUST SEARCH F / SET I MARK INDEX FSPØ7Ø, LDA I 7 ADM MARK XSK MARK	AT TRY / WAS THERE A FILE BY THIS NAME / YES - GO USE THAT ENTRY FOR EMPTY INDEX SPACE / INITIALIZE MARK POINTER / BUMP INDEX INDEX
451 0210 452 6525 453 0070 454 1000 455 1020 456 0007 457 1140 460 0010 461 0210	<pre>/ THERE IS SPACE A / FSPØ6Ø, XSK MARK JMP FSP1ØØ / / WE MUST SEARCH F / SET I MARK INDEX FSPØ7Ø, LDA I 7 ADM MARK XSK MARK</pre>	 AT TRY / WAS THERE A FILE BY THIS NAME / YES - GO USE THAT ENTRY FOR EMPTY INDEX SPACE / INITIALIZE MARK POINTER / BUMP INDEX INDEX
451 0210 452 6525 453 0070 454 1000 455 1020 456 0007 457 1140 460 0010 461 0210	/ FSPØ6Ø, XSK MARK JMP FSP1ØØ / WE MUST SEARCH F / SET I MARK INDEX FSPØ7Ø, LDA I 7 ADM MARK XSK MARK	/ WAS THERE A FILE BY THIS NAME / YES - GO USE THAT ENTRY FOR EMPTY INDEX SPACE / INITIALIZE MARK POINTER / BUMP INDEX INDEX
451 0210 452 6525 453 0070 454 1000 455 1020 456 0007 457 1140 460 0010 461 0210	FSP060, XSK MARK JMP FSP100 / WE MUST SEARCH F / SET I MARK INDEX FSP070, LDA I 7 ADM MARK XSK MARK	 / WAS THERE A FILE BY THIS NAME / YES - GO USE THAT ENTRY FOR EMPTY INDEX SPACE / INITIALIZE MARK POINTER / BUMP INDEX INDEX
452 6525 453 0070 454 1000 455 1020 456 0007 457 1140 460 0010 461 0210	JMP FSP100 WE MUST SEARCH F SET I MARK INDEX FSP070, LDA I 7 ADM MARK XSK MARK	/ YES - GO USE THAT ENTRY FOR EMPTY INDEX SPACE / INITIALIZE MARK POINTER / BUMP INDEX INDEX
453 0070 454 1000 455 1020 456 0007 457 1140 460 0010 461 0210	/ WE MUST SEARCH F / SET I MARK INDEX FSPØ7Ø, LDA I 7 ADM MARK XSK MARK	FOR EMPTY INDEX SPACE / INITIALIZE MARK POINTER / BUMP INDEX INDEX
453 0070 454 1000 455 1020 456 0007 457 1140 460 0010 461 0210	/ WE MUST SEARCH F / SET I MARK INDEX FSPØ7Ø, LDA I 7 ADM MARK XSK MARK	FOR EMPTY INDEX SPACE / INITIALIZE MARK POINTER / BUMP INDEX INDEX
453 0070 454 1000 455 1020 456 0007 457 1140 460 0010 461 0210	/ SET I MARK INDEX FSPØ7Ø, LDA I 7 ADM MARK XSK MARK	/ INITIALIZE MARK POINTER / Bump index index
453 ØØ7Ø 454 10ØØ 455 102Ø 456 ØØ07 457 114Ø 460 ØØ1Ø 461 Ø21Ø	SET I MARK INDEX FSPØ7Ø, LDA I 7 ADM MARK XSK MARK	/ INITIALIZE MARK POINTER / BUMP INDEX INDEX
454 1000 455 1020 456 0007 457 1140 460 0010 461 0210	INDEX FSPØ7Ø, LDA I 7 ADM MARK XSK MARK	✓ BUMP INDEX INDEX
455 1020 456 0007 457 1140 460 0010 461 0210	FSPØ7Ø, LDA I 7 adm mark XSK mark	/ BUMP INDEX INDEX
456 ØØØ7 457 1140 460 ØØ10 461 Ø210	7 ADM MARK XSK MARK	
457 1140 460 0010 461 0210	ADM MARK XSK MARK	
460 0010 461 0210	MARK XSK MARK	
461 0210	XSK MARK	
		/ TEST FOR END OF INDEX
462 Ø467	SKP	/ NOT END
463 6543	JMP RTRNØ	/ NO SPACE IN INDEX RETURN ZERO
464 1030	LDA I MARK	/ FIRST WORD OF INDEX NAME
465 146Ø	SAE I	/ CHECK FOR EMPTY ENTRY
466 5757	- 5757	
467 6455	JMP FSPØ7Ø	/ NOT EMPTY, TRY NEXT
	/	
	/ MARK POINTS TO A	AN UNUSED INDEX ENTRY
	/ PUT THE NAME INT	TO IT
	1	
470 1003	LDA PARM2	/ FIRST WORD OF USER NAME
471 1050	STA MARK	/ PLUNK INTO INDEX
472 1023	LDA I PARM2	/ 2ND
473 1070	STA I MARK	
474 1023	LDA I PARM2	/ 3RD
475 107Ø	STA I MARK	· · · ·
476 1023	LDA I PARM2	/ 4TH
477 1 <i>0</i> 70	STA I MARK	
500 1023	LDA I PARM2	/ TYPE CONTROL (S OR B)
501 1420	SHD I	/ IS TYPE S
502 2300	2300	
503 6510	JMP FSPØ8Ø	/ YES
504 1420	SHD I	/ IS TYPE B
505 0200	0200	
506 6521	JMP FSP090	/ YES
507 6543	JMP RTRNØ	/ NO BOMB NOW
	EJECT	
	ρq	
	The former and the	
	$\begin{array}{rrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrrr$	/ MARK POINTS TO / PUT THE NAME IN / 470 1003 LDA PARM2 471 1050 STA MARK 472 1023 LDA I PARM2 473 1070 STA I MARK 474 1023 LDA I PARM2 475 1070 STA I MARK 476 1023 LDA I PARM2 477 1070 STA I MARK 500 1023 LDA I PARM2 501 1420 SHD I 502 2300 2300 503 6510 JMP FSP080 504 1420 SHD I 505 0200 0200 506 6521 JMP FSP090 507 6543 JMP RTRN0 EJECT

0724			HE WANTS A SOURCE FILE SCRATCH THE BINARY POINTERS
0725	0510	0047	/ CEPARA SET LOO / TEMP POINTER
0/20 0777	0511	0010	MADE TENTIONIC
0121	0513	0010	
0730	0512	0221	XSK I LFZ / JANI SKIF
0/31	0510	0221	
0/32	0514	1020	
0/33	0515	5/5/	
0/34	0516	106/	STAT LP2 / STORE THEMAN
0/35	051/	106/	STAT LP2 7 IN BINARY PUINTERS
Ø736	Ø52Ø	6525	JMP FSP100
Ø737			
Ø74Ø			/ HE WANTS BINARY FILE
Ø741			
Ø742	Ø521	1020	FSP090, LDA I
Ø743	Ø522	5757	5757
Ø744	Ø523	107Ø	STA I MARK / STORE 57S
Ø745	Ø524	1070	STA I MARK /IN SOURCE POINTERS
0746			
Ø747			MARK NOW POINTS TO POINTER AREA OF DESIRED TYPE IN A NAMED INDEX ENTRY
Ø75Ø			
Ø751	Ø525	1000	FSP100, LDA / PICK UP STARTING BLOCK
0752	Ø526	0635	TRY
Ø753	Ø527	107Ø	STA I MARK / STORE IN INDEX
0754	Ø53Ø	1046	STA LP1 / STORE IN USERS LIST
Ø755	Ø531	1026	LDA I LP1 / GET LENGTH
Ø756	Ø532	1070	STA I MARK / STORE IN INDEX
Ø757			
Ø76Ø			NOW EVERYBODY IS HAPPY EXCEPT THE TAPE, WHICH HASNT BEEN UPDATED
Ø761			
Ø762	Ø533	ØØ62	SET I RDWR / SETUP WRITE CODE
Ø763	Ø534	-0704	704
0764	Ø535	0077	SET I RETURN / SETUP RETURN JMP
Ø765	Ø536	6542	JMP FSP110 / INST MOVED FOR LATER USE
0766	Ø537	0061	SET I PARAM / SETUP PARAMETER POINTER
Ø767	Ø54Ø	Ø277	XPARM
Ø77Ø	0541	6Ø44	JMP RWENT / GO DO THE WRITE
Ø771	Ø542	Ø235	FSP11Ø, XSK I RET3 / BUMP RETURN ADDR
Ø772	Ø543	0040	RTRNØ, SET Ø / MOVE RETURN JMP TO Ø
0773	0544	ØØ15	RET3
Ø774	Ø545	Ø5ØØ	IOB
Ø775	Ø546	6244	LRMF / RESTORE USERS FIELDS
Ø776	Ø547	6000	JMP Ø / GO TO CALLER
Ø777			EJECT

P.10

	1000			/		
	1001			·/ · F	IND POSSIBLE (CONFLICT BETWEEN INDEX ENTRY AND TRIAL STARLING BLOCK
	1002 _			1		
	1003			/ 0	ALLER MUST SET	T OR CLEAR I-BI STORE FOR TO SELECT DESIRED FILE AREA
	1274			1		
	1005	0550	0056	ECENNO.	SET RET2	/ SAVE RETURN
	1226	0551	99999		2	
	1007	2552	0000		SET I VENT	A THITTALTRE DOTHTED TO THOSE IN CORE
	1007	0557	1004		TNOEVIA -	/ INTIALIZE FOINTER TO INDEX IN CONC
	1010	0555	1000	505340	INDEXTO	CHID ON END OF INDEX
	1011	0554	0224	FUFDID,	ASN I APINI .	/ SKIP ON END OF INDEX
	1012	0555	040/		SKP	
	1013	0556	6016		JMP RE12	/ END OF INDEX, RETURN NO CONFLICT
	1014	0557	Ø224		XSK I XPNI	/ INCREMENT AGAIN, NO SKIP POSSIBLE
	1015	0560	1000		LDA	/ PICK UP POINTER
	1016	0561	0004		XPNT	
	1017	Ø562	Ø323		ROR I 3	/ MOVE BIT 9 TO LINK
	1020	Ø563	0452		LZE	/ SKIP IF NAME AREA OF ENTRY
	1021	0564	6602		JMP FCF040	/ JMP IF POINTER AREA
	1022	0565	1004		LDA XPNT	/ GET FIRST WORD OF NAME
	1023	0566	1460		SAF I	/ SKIP IF EMPTY ENTRY
	1024	0567	5757		5757	
	1025	0570	· 6576		INP FOFATA	/ WORD IS VALID NAME
	1026	0571	1020			/ CET CONSTANT 6
	1020	0572	1020		LUA I	/ GET CONSTANT S
	1027	0572	1140		0	ADDDERS NEXT ENTRY 2. THIS ONE IS EMDIV
	1030	0573	1140		AUM	AUDRESS NEXT ENTRY - 2, THIS UNE IS EMPTT
	1031	05/4	0004		XPNI	
	1032	0575	6554		JMP FCF010	Z TRY NEXT INDEX ENTRY
	1033	Ø576	1020	FCFØ30,	LDA I	/ INCREMENT BY 4
	1034	Ø577	0004		4	
	1035	Ø6 ØØ	1140		ADM	ZTO ADDRESS POINTER
	1036	Ø6Ø1	0004		XPNT	
	1037		8 - C	· /		
	1040	1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 - 1997 -		/ X	PNT NOW ADDRES	SSES A STARTING BLOCK NO
	1041			/		
u.	1042	0602	1004	FCFØ4Ø,	LDA XPNT	/ PICK UP STARTING BLOCK
	1043	Ø6Ø3	112Ø		ADA I	/ SUBTRACT INDEX LOCATION
	1044	0604	7430		-347	
	1045	0605	0451	FCFØ5Ø,	APO	/ REVERSE SENSE BIT MAY BE SET BY CALLER
	1046	0606	6554		JMP FCF010	/ TRY AGAIN IF WRONG FILE AREA
	1047	0607	0045		SET XPNT2	/ TEMP POINT TO LENGTH WORD
	1050	0610	0004		YPNT	
	1051	0611	1025		INA T YONT?	A PTOK HD LENGTH
	1052	0612	0451		LUA I AFNIZ	/ SKID IE LENGTH DOS
	1057	0012	6554		ND ECEAIA	
	1050	0010	0))4		UMP FURDIO	/ NEG LEN - NO FILE HERE
	1024				EJELI	
	-					
				$ \land \land \land \land$	1	
		1		F.I	1	
				· · ·		

	1056			WE NOW HAT	E A VALID	INDEX ENTRY	12	
	1057			COMPARE T	TO TRY		/	
· •	1060			/				
	1061	0614	1004	IDA XPN	r /	STARTING BLOCK OF THIS FILE		
	1042	0615	1540			CLEAD CARRACE IN HIGH THREE BITS		
	1002	0616	7000	7000	,	CLEAR GARDAGE IN DIGA PRACE BY C		
	1003	0100	7000	7004				
	1004	001/	0017	LUM IDO TON		COURTRACT VOTART FROM TRY		
	1065	0020	2035	AUU IRY		SUBIRACI XSTART FRUM IRT		
	1066	0621	0451	APO	. /	SKIP IF IRY ABOVE XSTART		
	1067	Ø622	663Ø	JMP FCF	360 /	JMP IF BELOW		
	1070							
	1071			/ TRY IS ABO	DVE THE ST	ART OF THIS FILE		
	1072			/				
	1073	0623	0017	COM	1	MAKE DIFFERENCE NEGATIVE		
	1074	0624	1105	ANA XPN	2 /	SUBTRACT DIFFERENCE FROM FILE LENGTH		
	1075	Ø625	0451	APO	- /	FERD RESULT WILL BE NEG (7777)		
	1075	0626	6554		1 0 /	NO CONFLICT. TRY NEXT ENTRY		
	1070	0020	4477			CONFLICT FOUND - DETUDN		
	10//	0021	0033		0/10 /	CUNFLICT FOUND RETORN		
	1100							
	1101			/ THE TRY 1:	S BELOM OR	AT THE START OF THIS FILE		
	1103	0630	2636	CE060. ADD TRY	EN /	SUBTRACT STARTING DIFF FROM LEN OF TRY		
	1104	0631	Ø451	AP0		SKIP IF CONFLICT		
	1104	0601	6557	IND ERE	1 a /	NO CONFLICT - TRY NEXT ENTRY		
	1100	0002	0004	JINE EGEN	10 /	NO COMPLICE THE NEXT ENDS		
	1100					ELICT - DETHON IT TO CALLED		
	110/			WE HAVE FU	JUND A CON	IFLICE RETURN IF TO CALLER		
	1110	~ < 77	007/			THEREMENT RETURN ADDRESS		
	1111	0633	0236	CF0/0, XSK 1 RE	12 /	INCREMENT REFORM ADDRESS		
	1112	0634	6016	JMP REI	2	GO BACK		
	1113							
	1114			WORK AREA				
	1115		1. Contraction (1997)	/				
	1116	Ø635	ØØØØ	rry, Ø				
	1117	Ø636	0000	TRYLEN, Ø		0		
	1120	0637	0000	SVTRY, Ø	PI	1		
	1121	Ø64Ø	ØØØØ	JPLIM, Ø	I ()	•		
	1122			FJECT				
	-							
			• •					

1122 SAME CALLING SEQUENCE AS LOOK EXCEPT NO ALTERNATE RETENS 1121 F642 (0055) DEL000, SET RET3 / SAVE RETURN 1131 F642 (0055) DEL000, SET RET3 / SAVE RETURN 1132 F643 (045) JMP LKP200 / LDDKUP NAME IN INDEX 1133 F644 (055) JMP RTRNN / DDM TFIND THE FILE 1134 F645 (053) JMP RTRNN / DDM TFIND THE FILE 1135 F644 (052) LDA I / DECREMENT XPAT2 1136 R647 (776) -1 TO ADDRESS START BLOCK FIELD 1136 R647 (776) LOA / EMPTY AREA INDICATOR 1144 R655 (056) STA XPAT2 / ZAP START BLOCK FIELD 1144 R655 (056) STA XPAT2 / DTTO LTN 1146 R657 (026) STA XPAT2 / DTTO LTN 1147 R668 (060) STA XPAT2 / DTTA FLATYBE ENTER 1147 R668 (060) STA XPAT2 / DTTA FLATYBE ENTER ILE 1147 R668 (060) LOA / SET PTO THEFT TYPE ENTER ILE 1148 R667 (060) R600 LOA / SET PTO THEFT TYPE	1125			, ,	LLETE REHOVE	M F	TLE FROM THE DIAL INDEX
1132 2641 265 SET RET3 / SAVE RETURN 1131 2642 1000 2 JPP LRP202 / LOCKJP NAME IN IADEX 1132 2643 6144 JPP LRP202 / LOCKJP NAME IN IADEX 1134 2644 643 JPP LRP202 / LOCKJP NAME IN IADEX 1135 1646 1022 LD1 / DECREMENT XPNT2 1136 1646 1022 LD1 / DECREMENT XPNT2 1137 2657 144 ADP /TO ADDRESS START BLOCK FIFLD 1148 2653 1148 ADP /TO ADDRESS START BLOCK FIFLD 1149 2654 1445 STA XPNT2 / PAP START BLOCK FIFLD 1149 2655 STA XPNT2 / ADDRESS START BLOCK FIFLD 1144 2655 STA XPNT2 / PAP START BLOCK FIFLD 1144 2656 BOED VANTE ADP TOTOLEN 1144 2656 STA XPNT2 / PAP START BLOCK FIFLD 1154 2666 STA XPNT2 / ADD TOTOLEN 1154 2666 STA XPNT2 / PAP START BLOCK FIFLD 1155 <td>1126</td> <td></td> <td></td> <td>/. / S</td> <td>AME CALLING SEQU</td> <td>JENC</td> <td>F AS LOOK EXCEPT NO ALTERNATE REFURNS</td>	1126			/. / S	AME CALLING SEQU	JENC	F AS LOOK EXCEPT NO ALTERNATE REFURNS
133 Cod1 (005) DEL000, SET RET3 / SAVE RETURN 133 Cod2 (002) UP L4P202 / LOCKUP NAME IN INDEX 133 Cod3 (004) UP RTENA / DIDAT TYD UHE FILE 134 Cod4 (053) UP RTENA / DIDAT TYD UHE FILE 135 Cod4 (053) UP RTENA / DIDAT TYD UHE FILE 134 Cod4 (053) UP RTENA / DIDAT TYD UHE FILE 135 Cod4 (053) UP RTENA / DIDAT TYD UHE FILE 134 Cod5 (004) APRT2 / JOADRESS START BLOCK FIELD 1141 Cod5 (004) Cod APRT2 / ZAP START BLOCK FIELD 1142 Cod5 (004) STA XPNT2 / ZAP START BLOCK FIELD IIIIII 1144 Cod5 (004) STA XPNT2 / ZAP START BLOCK FIELD 1144 Cod5 (004) STA XPNT2 / PAP START BLOCK FIELD 1144 Cod6 (006) Cod Cod XPNT2 / PAP START BLOCK FIELD 1144 Cod6 (006) Cod Cod XPNT2 / PAP START BLOCK FIELD 1144 Cod6 (006) Cod XPNT2 / PAP START BLOCK FIELD 1144 Cod6 (006)	1127			/			
1131 Cr642 CR20 C 1132 Cr644 JMP RTRW JICNT FIND THE FILE 1134 Cr645 Cr645 JMP RTRW JICNT FIND THE FILE 1135 Cr646 L024 LDA J JICNT FIND THE FILE 1135 Cr646 L024 LDA J JICNT FIND THE FILE 1135 Cr646 L024 LDA J JICNT FIND THE FILE 1136 Cr647 TR77 T JINT 1137 Cr635 JINT FIND THE FILE JICNT FIND THE FILE 1144 Cr635 JINT FIND THE FILE JINT 1145 Cr636 JINT ZET FINT JINT 1146 Cr635 JINT JINT ZET FINT JINT 1145 Cr636 JINT JINT ZET FINT JINT 1145 Cr636 JINT JINT JINT JINT JINT	1130	2641	0055	DEL000,	SET RET3	1	SAVE RETURN
1132 C+4-5 6144 JMP LRP322 / DOCUP LANK IN INDEX 1134 C+4-5 6543 JMP RTRN2 / DICH FIND THE FILE 1134 C+4-5 6543 JMP RTRN2 / DICH FIND THE FILE 1135 C+4-5 644 1201 / DECREMENT XPAT2 1136 C+4-7 776 -1 ////////////////////////////////////	1131	2642	0000		Ø		
1133 U=44 0:043 UP RTRW2 / DIONT FIND THE FILE 1134 U=46 1020 LDA [/ DECREMENT XPNT2 1135 U=46 1020 LDA [/ DECREMENT XPNT2 1136 U=46 1020 LDA [/ DECREMENT XPNT2 1137 U=46 1020 LDA [/ DECREMENT XPNT2 1140 U=50 U=46 NUT PERTURE NUT 1141 U=55 U=46 STA XPNT2 / Z=P START BLOCK FIELD 1142 U=56 STA XPNT2 / Z=P START BLOCK FIELD 1143 U=46 U=46 STA XPNT2 / Z=P START BLOCK FIELD 1144 U=55 STA XPNT2 / Z=P START BLOCK FIELD 1144 U=455 U=46 STA XPNT2 / Z=P START BLOCK FIELD 1144 U=455 U=46 STA XPNT2 / Z=P START BLOCK FIELD 1144 U=455 U=46 STA XPNT2 / Z=P START BLOCK FIELD 1157 U=462 W=462 STC XPNT2 / I=100 1158 U=464 STA XPNT / X=P TOTHER TYPE EMPTY IIEE<	1132	0643	6144		JMP LKPØØØ	/	LOOKUP NAME IN INDEX
1134 P645 6543 JMP RTRVA / DECREMENT XPAT2 1135 P646 122 LDA I / DECREMENT XPAT2 1136 P647 7776 -1 1137 P651 P489 LLA / DECREMENT XPAT2 1141 B652 1827 LDA I / EMPTY AREA INDICATOR 1144 B653 P655 STA I XPNT2 / ZAP STAFT BLOCK FIELD 1144 B653 P655 STA I XPNT2 / ADDRESS OTHER TYPE LEN FIELD 1146 B657 P868 RAD / OTTER 1147 P660 B600 I XPNT2 / ADDRESS OTHER TYPE LEN FIELD 1158 P664 P872 STC XPNT2 / FICK LENGTH OF DTHER-TYPE FILE 1159 P664 P872 STC XPNT2 / FICK LENGTH OF DTHER-TYPE FILE 1150 P664 P872 STC XPNT2 / FICK LENGTH OF DTHER-TYPE FILE 1152 P664 P874 APO I / STE XPNT / EMPTY 1154 P665 STC XPNT / EMP FIRST WORD OF NAME ////////////////////////////////////	1133	0644	6543		JMP RTRNØ	/	DIDNT FIND THE FILE
1135 P246 1928 LDA 1 / DECREMENT XPMT2 1136 0647 776 -1 1137 0658 1140 ADM TO ADDRESS START BLOCK FIELD 1140 P2651 1020 LOA I / EMPTY AREA INDICATOR 1141 0653 1020 LOA I / EMPTY AREA INDICATOR 1142 0655 STA PMT2 ZAP START BLOCK FIELD 1143 0654 1045 STA PMT2 ZAP START BLOCK FIELD 1144 0655 STA I XPNZ ZAP START BLOCK FIELD 1144 0655 STA PMT2 / ADDRESS START BLOCK FIELD 1144 0655 STA PMT2 / ADDRESS START BLOCK FIELD 1144 0656 STA PMT2 / ADDRESS START BLOCK FIELD 1144 0656 STA PMT2 / ADDRESS START BLOCK FIELD 1145 0656 STAPT ZAP START BLOCK FIELD 1157 0661 0607 XPMT2 1158 0664 471 APD T / STAPTY 1158 0666 1028 LDA I / STAPTY 1158	1134	@645	6543		JMP RTRNØ	/	DITTO
1136 0647 7776 -1 1137 0658 1140 ADH /TD ADDRESS START BLOCK FIELD 1140 0651 0627 LOA I / EMPTY AREA INDICATOR 1141 0652 1282 LOA I / EMPTY AREA INDICATOR 1142 0655 STA X PNT2 ZAP START BLOCK FIELD 1144 0656 1065 STA X PNT2 ZAP START BLOCK FIELD 1144 0657 000 LDA / GET POINTER 1145 0668 000 DA / GET POINTER 1146 0657 0005 XPNT2 / IDTO LEN 1147 0668 0020 Z / ADDRESS OTHER TYPE LEN FIELD 1150 0661 0027 YPNT2 / IDTO LEN 1151 0666 0128 JMM PD I / SKIP F OTHER-TYPE EMPTY CLEAR NAME 1151 0666 0129 JDT / BDT / SKIP F OTHER TYPE EMPTY CLEAR NAME 1152 0666 0129 JDAT / YDAT / SKIP FITE INDEX / LEAR NAME 1153 1064 STA I XPNT / ADD <td>1135</td> <td>0646</td> <td>1020</td> <td></td> <td>LDA I</td> <td>1</td> <td>DECREMENT XPNT2</td>	1135	0646	1020		LDA I	1	DECREMENT XPNT2
1137 r656 1140 ADM /TO ADDRESS START BLOCK FIELD 1140 r651 0055 XPN12 1141 r652 1220 LOA I / EMPTY AREA INDICATOR 1142 r653 5757 5757 1143 r654 1245 STA YENT2 ZAP START BLOCK FIELD 1144 r655 1265 STA YENT2 ZAP START BLOCK FIELD 1144 r655 1265 STA YENT2 ZAP START BLOCK FIELD 1144 r656 1268 LOA / GET POINTER 1145 r6661 8282 Z 1151 r6661 8282 Z 1152 r6661 828 Z 1153 r6661 8282 Z 1154 r665 ritu JAPD I / SKIP IF OTHER-TYPE FILE 1154 r6665 ritu JAPD I / SKIP IF OTHER-TYPE FENTY 1154 r6667 ritu JAPD I / SKIP IF OTHER-TYPE FENTY 1156 ritu ritu ritu SKIP IF OTHER-TYPE FENTY 1160 ritu </td <td>1136</td> <td>0647</td> <td>7776</td> <td></td> <td>-1 .</td> <td></td> <td></td>	1136	0647	7776		-1 .		
1141 0651 0005 XPNT2 1141 0652 1622 LOA I / EMPTY AREA INDICATOR 1142 0653 1645 STA XPNT2 ZAP START BLOCK FIELD 1144 0655 1865 STA XPNT2 ZAP START BLOCK FIELD 1144 0655 1865 STA XPNT2 ZAP START BLOCK FIELD 1145 0656 1806 DA / GET POINTER 1146 0657 8005 XPNT2 1147 0664 0802 2 1151 0664 1802 2 1152 0664 100 JAP DICLOSO / FOR HARTYPE EMPTY FILE 1153 0664 070 JAP DICLOSO / GC RE-HRITE INDEX 1154 0666 170 JAP DICLOSO / GC RE-HRITE INDEX 1155 0666 1804 STA XPNT / ZAP FIRST WORD OF NAME 1156 0667 1804 STA XPNT / ZAP FIRST WORD OF NAME 1161 0671 1804 STA XPNT / ZAP FIRST WORD OF NAME 1162 0673 1804 STA XPNT <	1137	0650	1140		ADM	1.	TO ADDRESS START BLOCK FIELD
1142 0652 1020 LDA I / EMPTY AREA INDICATOR 1142 0655 1020 LDA I / EMPTY AREA INDICATOR 1144 0655 1020 STA I XMT2 / DITTO LEN 1145 0656 0120 LDA / GET POINTER 1146 0657 0002 Z 1151 0661 0002 Z 1151 0661 0002 Z 1152 0663 1020 LDA YMT2 / FICK LENGTH OF DITHER-TYPE FILE 1153 0664 0075 CONSTRUCT 1154 0656 0710 JMP DEL030 / GET-RATTE INDEX 1155 0666 1020 LDA I / SKIP IF OTHER-TYPE EMPTY 1156 0667 5775 5757 1157 0678 1044 STA I XPNT / ZAP FIRST WORD OF NAME 1160 0677 1054 STA I XPNT / ZAP FIRST WORD OF NAME 1161 0672 1064 STA I XPNT / ZAP 1162 0673 1024 STA I XPNT / ZAP 1164 STA I XPNT / SRD 1167 0678 0073 1064 STA I XPNT / SRD 1168 0677 1075 STA I XPNT / SRD 1169 0675 0757 JT57 1167 0678 1044 STA I XPNT / SRD 1160 0677 1073 LO44 STA I XPNT / SRD 1161 0672 1064 STA I XPNT / SRD 1162 0673 1024 STA I XPNT / SRD 1164 STA I XPNT / SRD 1165 0676 073 LO44 STA I XPNT / SRD 1166 0677 6710 JMP DEL030 / YES - REMRITE 1167 // MAS THAT THE LAST FILE 1168 // MAS THAT THE LAST FILE 1169 071 1064 STA I XPNT / POINT TO START OF INDEX 1169 0675 077 T NDEL030 / YES - REMRITE 1160 0676 1073 LO44 STA I XPNT / POINT TO START OF INDEX 1167 073 1064 STA I XPNT / POINT TO INDEX 1169 0675 0710 JMP DEL030 / YES - REMRITE 1170 0761 1464 DEL0310 / YES - REMRITE 1171 0770 0780 0260 XSK XPNT / CHECK FOR END 1172 0780 0260 XSK XPNT / CHECK FOR END 1173 0780 0264 SET I XPNT / POINT TO INDEX 1174 // INDEX IS EMPTY: MAKE IT A NON-INDEX 1175 // RE-ARITE INDEX / NOT YET - LODP // INDEX IS EMPTY: MAKE IT A NON-INDEX 1280 2786 0264 SET I XPNT / LAST ONE ? 1297 0765 // MAE ARITE INDEX / NOE // N	1140	Ø651	UØØ5		XPNT2		
1142 0653 5757 5757 72 72 72 72 72 72 72 72 72 72 72 72 72	1141	0652	1020		LDA I	/	EMPTY AREA INDICATOR
1143 0654 1045 STA XPNT2 / ZAP START BLOCK FIELD 1144 0655 1080 LDA / GET POINTER 1145 0656 1680 BCD 1 / ADDRESS OTHER TYPE LEN FIELD 1146 0657 0405 XPNT2 1151 0661 0402 2 1152 0663 1025 LDA XPNT2 / PICK LENGTH DF DTHER-TYPE FILE 1153 0664 0471 APD I / SKIP IF OTHER-TYPE FILE 1154 0665 0710 LPM PEL202 / GO RE-WRITE INDEX 1154 0665 0720 LDA XPNT2 / PICK LENGTH DF OTHER-TYPE FILE 1154 0665 0710 LPM PEL202 / GO RE-WRITE INDEX 1156 0667 0757 7777 / SOT 1160 0671 1064 STA I XPNT / ZPF FIRST WORD OF NAME 1161 0672 1264 STA I XPNT / ZPF FIRST WORD OF NAME 1162 0673 1264 STA I XPNT / ZPF FIRST WORD OF NAME 1164	1142	0653	5757		5757		
1144 0655 1065 STA I XPNT2 / DITTO LEN 1146 0657 0005 XPNT2 1147 0668 1660 BCO I / ADDRESS OTHER TYPE LEN FIELD 1150 0661 0002 2 1151 0662 4005 STC XPNT2 1152 0663 1005 LDA XPNT2 / PICK LENGTH OF OTHER-TYPE FILE 1153 0664 0471 APO I / STN I / PICK LENGTH OF OTHER-TYPE FILE 1154 0665 6710 JMP DEL030 / GO RE-WRITE INDEX 1155 0666 120 LDA I / BOTH TYPES EMPTY CLEAR NAME 1156 0666 7575 7577 1157 0673 1044 STA I XPNT / ZAP FIRST HORD OF NAME 1160 0671 1064 STA I XPNT / ZAP FIRST HORD OF NAME 1161 0672 1064 STA I XPNT / ZAP FIRST HORD OF NAME 1162 0673 1064 STA I XPNT / ZAP 1164 / WAS THAT THE LAST FILE 1166 0674 0064 SET I XPNT / YES - REMRITE 1170 0675 0777 INDEX-1 1170 0676 1464 DEL010 SAE I XPNT / YES - REMRITE 1172 0701 1667 JMP DEL030 / NOT YET - LOOP 1174 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1173 0701 6676 JMP DEL010 / NOT YET - LOOP 1174 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1176 077 10 CLR 1177 0702 0011 CLR 1280 2703 0044 STA I XPNT / YES - REMRITE 1172 2700 0204 SIS I XPNT / YES - REMRITE 1173 0701 6676 JMP DEL010 / NOT YET - LOOP 1174 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1175 // INDEX SIS I XPNT / VERS NOR 1176 // INDEX IS EMPTY: MAKE IT A NON-INDEX 1177 2702 0011 CLR 1280 2703 0044 SIA I XPNT / YERO NOE NOR 1174 // INDEX IS EMPTY: MAKE IT A NON-INDEX 1175 // RE-WRITE INDEX 1176 // RE-WRITE INDEX 1177 2700 0055 JMP2 / NO - LOOP 1174 // RE-WRITE INDEX 1174 // RE-WRITE INDEX 1175 // RE-WRITE INDEX 1175 // RE-WRITE INDEX 1176 // RE-WRITE INDEX 1176 // RE-WRITE INDEX 1177 2700 0054 SIA I RETURN / SET WRITE CODE 1284 2777 GF65 JMP2 / NO - LOOP 1284 2777 GF65 JMP2 / NO - LOOP 1285 // RE-WRITE INDEX 1295 // RE-WRITE INDEX 1296 // RE-WRITE INDEX 1296 // RE-WRITE INDEX 1296 // RE-WRITE INDEX 1297 // RE-WRITE INDEX 1297 // RE-WRITE INDEX 1296 // RE-WRITE INDEX 1296 // RE-WRITE INDEX 1297 // RE-WRITE INDEX 1297 // RE-WRITE INDEX 1297 // RE-WRITE INDEX 1299 // RE-WRITE INDEX 1299 // RE-WRITE INDEX 1299 // RE-WRITE INDEX 1299 // RE-WRITE // DO T	1143	0654	1045		STA XPNT2	/	ZAP START BLOCK FIELD
1145 0657 0800 LDA / GET POINTER 1146 0657 0805 XPNT2 1147 0668 1002 2 1151 0664 0802 2 1152 0663 1002 2 1151 0664 0802 2 1152 0663 1005 LDA XPNT2 / PICK LENGTH OF OTHER-TYPE EMPTY 1154 0665 0710 JMP DEL030 / OR R=wRITE INDEX 1156 0665 1020 LDA I / SNIP IF OTHER-TYPE EMPTY CLEAR NAME 1156 0667 5757 7577 / BOTH TYPES EMPTY CLEAR NAME 1156 0667 1064 STA I XPNT / ZAP FIRST WORD OF NAME 1160 0671 1064 STA I XPNT / ZAP 1161 0672 1064 STA I XPNT / SAD 1162 0673 1064 STA I XPNT / SAD 1162 0673 1064 STA I XPNT / SAD 1162 0673 1064 STA I XPNT / SAD 1166 0674	1144	0655	1065		STA I XPNT2	/	DITTO LEN
1146 0657 0005 XPN12 1150 0664 0002 2 1151 0664 0002 2 1152 0663 1005 LDA XPN12 / PICK LENGTH DF DTHEN-TYPE FILE 1153 0664 0471 APO I / SKIP IF OTHER-TYPE FILE 1153 0664 0471 APO I / SKIP IF OTHER-TYPE FILE 1154 0665 6710 JMP DEL030 / GC RE-WRITE INDEX 1155 0666 1020 LDA I / SOTH TYPES EMPTY CLEAR NAME 1157 0670 1024 STA YNT / ZAP FIRST HORD OF NAME 1160 0671 1064 STA I XPNT / ZAP FIRST HORD OF NAME 1161 0672 1064 STA I XPNT / ZAP 1162 0673 1064 STA I XPNT / ZAP 1164 Y / WAS THAT THE LAST FILE 1165 / 1166 0674 0064 SET I XPNT / POINT TO START OF INDEX 1167 0675 0777 INDEX-1 1170 0676 0777 INDEX-1 1170 0676 0777 INDEX-1 1171 0676 1074 JMP DEL030 / YES - REWRITE ENTRY 1172 0701 0676 JMP DEL030 / YES - REWRITE 1173 0701 0676 JMP DEL030 / YES - REWRITE 1174 077 1 INDEX-1 1175 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1176 077 INDEX-1 1177 0702 0011 CLR 1280 2783 1084 STA I XPNT / POINT TO INDEX 1176 // INDEX IS EMPTY: MAKE IT A NON-INDEX 1177 0702 0011 CLR 1280 2785 1084 STA I XPNT / POINT TO INDEX 1290 2785 1084 STA I XPNT / POINT TO INDEX 1201 2784 3777 INDEX-1 1202 2785 1084 STA I XPNT / POINT TO INDEX 1203 2785 1084 STA I XPNT / POINT TO INDEX 1204 2785 1084 STA I XPNT / POINT TO INDEX 1205 / RE-WRITE INDEX / 1224 2727 6785 JMP2 / NO - LOOP 1225 / RE-WRITE INDEX 1224 2727 6785 JMP2 / NO - LOOP 1225 // RE-WRITE INDEX 1226 2745 1084 STA I XPNT / SET WRITE CODE 1227 // RE-WRITE INDEX 1229 2747 328 2284 STA I XPNT / SET WRITE CODE 1224 2727 6785 JMP2 / NO - LOOP 1225 // RE-WRITE INDEX 1226 // RE-WRITE INDEX 1227 271 272 XPARM 1228 2716 0344 STA I YPARM 1229 2716 0344 STA I YPARM 1220 2716 0344 STA I YPARM 1221 2712 277 XPARM 1222 2716 0344 STA I YPARM 1223 2716 0344 STA I YPARM 1224 2716 0344 STA I YPARM 1225 2715 // REAMITE INDEX // RE-WRITE YPARM // RE-WRITE INDEX // RE-WRITE INDEX // RE-WRITE YPARMENT / DO THE WRITE THING, AND RETUP, TO CALLEC	1145	0656	1000		LDA	/	GET POINTER
1147 0660 1660 000 2 1150 0661 0000 2 1151 0662 4005 STC XPNT2 1152 0663 1005 LDA XPNT2 / PICK LENGTH DF OTHER-TYPE FILE 1153 0664 0471 APO I / SKIP IF OTHER-TYPE FILE 1154 0665 6710 JMP DEL030 / SO RE-KRITE INDEX 1155 0666 1020 LDA I / BOTH TYPES EMPTY CLEAR NAME 1156 0667 5757 5757 1157 0670 1044 STA I XPNT / ZAP FIRST WORD OF NAME 1160 0671 1064 STA I XPNT / ZAD 1161 0672 1064 STA I XPNT / ATH 1162 0673 1064 STA I XPNT / ATH 1164 / WAS THAT THE LAST FILE 1165 0674 0864 SET I XPNT / POINT TO START OF INDEX 1166 0674 0864 SET I XPNT / YES - REWRITE 1170 0676 1464 DEL010, SAE I XPNT / CHECK FOR END 1172 0700 0204 XSK XPNT / CHECK FOR END 1173 0701 0676 JMP DEL030 / NOT YET - LOOP 1174 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1176 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1177 0702 0011 CLR 1178 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1179 0767 677 INDEX-1 1170 0770 JMP DEL010 / NOT YET - LOOP 1171 0777 INDEX-1 1172 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1174 // RE-WRITE INDEX 1260 // RE-WRITE INDEX 1274 0777 INDEX-1 1280 2785 0864 STA I XPNT / DEL00 P / RE-WRITE INDEX 1281 2776 785 JMP2 / NO - LOOP 1283 2786 0204 XSK XPNT / LAST ONE ? 1284 2787 6795 JMP2 / NO - LOOP 1285 // RE-WRITE INDEX 1295 1/ RE-WRITE INDEX 1296 2/10 3/27 SET I RETURA / RETURN JMP FOR R/K ROUTINE 1213 2/10 3764 SET I NOWA / SET WRITE CODE 1214 2/10 3764 SET I PARAM 1215 2/10 0344 JMP RENT / OD THE WRITE THING, AND RETUPN TO CALLER 1297 XP ARAM 1297 XPARAM 1297 XPARAM 1297 XPARAM 1297 XPARAM 1297 XPARAM	1146	0657	0005		XPNT2		
1150 0661 0002 2 1151 0662 1005 LDA XPNT2 / FICK LENGTH OF OTHER-TYPE FILE 1152 0663 1005 LDA XPNT2 / FICK LENGTH OF OTHER-TYPE FILE 1153 0664 0471 APO I / SKIP IF OTHER-TYPE EMPTY 1154 0666 120 LDA I / BOTH TYPES EMPTY CLEAR NAME 1156 0667 5757 5757 5757 1157 0678 1204 STA XPNT ZAP FIRST WORD OF NAME 1160 0671 1204 STA XPNT ZAP FIRST WORD OF NAME 1161 0672 1204 STA XPNT ZAP FIRST WORD OF NAME 1162 0673 1204 STA XPNT ZAP FIRST WORD OF NAME 1164 / WAS THAT THE LAST FILE / 1165 / / WAS THAT THE LAST FILE 1166 0674 0064 SET 1 XPNT / POINT TO START OF INDEX 1170 0676 1004 DEL010 / NOT YET - LOOP 1171 0676 JMP DEL010 / NOT YET - LOOP 1174 / <td>1147</td> <td>066Ø</td> <td>1660</td> <td></td> <td>BCO I</td> <td>1</td> <td>ADDRESS OTHER TYPE LEN FIELD</td>	1147	066Ø	1660		BCO I	1	ADDRESS OTHER TYPE LEN FIELD
1151 0662 4005 STC XPNT2 1152 0663 1205 LDA XPNT2 / PICK LENGTH OF OTHER-TYPE FILE 1153 0664 0471 APO I / SKIP IF OTHER-TYPE EMPTY 1154 0665 6700 LDA I / BOTH TYPES EMPTY CLEAR NAME 1156 0667 575 5757 1157 0670 1244 STA XPNT / ZAP FIRST WORD OF NAME 1160 0671 1264 STA I XPNT / ZAD 1161 0672 1264 STA I XPNT / ZAD 1162 0673 1264 STA I XPNT / ATH 1164 / WAS THAT THE LAST FILE 1165 // 1166 0674 0264 SET I XPNT / POINT TO START OF INDEX 1167 0676 1464 DEL010, SAE I XPNT / DINT TO START OF INDEX 1167 0676 1464 DEL010, SAE I XPNT / START OF AND-EMPTY ENTRY 1170 0676 1464 DEL010, SAE I XPNT / CHECK FOR END 1172 0700 0224 XSK XPNT / CHECK FOR END 1173 0701 0676 JMP DEL010 / NOT YET - LOOP 1174 // INDEX IS EMPTY: MAKE IT A NON-INDEX 1176 // INDEX IS EMPTY: MAKE IT A NON-INDEX 1177 0702 0011 CLR 1280 2783 0264 SET I XPNT / POINT TO INDEX 1290 777 INDEX-1 1202 2785 1064 SET I XPNT / POINT TO INDEX 1204 0777 INDEX-1 1204 0777 NADEX-1 1205 // 1205 // 1206 0765 JMP PEL010 / NOT YET - LOOP 1217 // CP02 0011 CLR 1224 0777 INDEX-1 1230 2785 DEL030 / SET I XPNT / POINT TO INDEX 1231 C744 0777 INDEX-1 1232 2785 DEL030 / SET I XPNT / POINT TO INDEX 1234 2786 0224 XSK XPNT / LAST ONE ? 1235 // RE-ARITE INDEX 1236 // RE-ARITE INDEX 1237 1064 SET I XPNT / POINT TO INDEX 1238 2786 DEL030 // SET I ROWA / SET WRITE CODE 1239 // RE-ARITE INDEX 1234 2785 504 SET I ROWA / SET WRITE CODE 1234 2785 504 SET I ROWA / SET WRITE CODE 1235 // SET I RELOVA / RETURN JUMP FOR R/K ROUTINE 1234 2710 0704 724 1234 2710 0704 SET I RELOVA / RETURN JUMP FOR R/K ROUTINE 1235 // SET I PARAM 1236 2716 0344 UP RTRV2 1234 2716 0344 UP RTRV2 1234 2716 0344 UP RTRV2 1235 // SET I PARAM 1236 2716 0344 UP RTRV2 1237 // SET I PARAM 1236 2716 0344 UP RTRV2	1150	0661	0002		2		
1152 0663 1005 LDA XPNT2 / PICK LENGTH OF DIHER-TYPE FILE 1153 0664 0471 APO I / SKIP IF OTHER-TYPE FILE 1154 0665 6710 JMP DEL030 / GO RE-WRITE INDEX 1155 0666 1020 LDA I / BOTH TYPES EMPTY CLEAR NAME 1156 0667 5757 5757 / STAT 1160 0671 1064 STA I XPNT ZAP FIRST HORD OF NAME 1161 0671 1064 STA I XPNT ZAP 1162 0673 1064 STA I XPNT ARD 1164 STA I XPNT ARD YAR YARD 1166 0674 0674 077 INDEX-1 POINT TO START OF INDEX 1167 0675 0777 INDEX-1 / IS THERE A NON-EMPTY ENTY ENTY 1170 0676 1464 DEL010 / YES - REWRITE ENTY 1171 0676 JMP DEL030 / YES - REWRITE ENTY ENTY 1177 0700 0204 XEK XPNT / CHECK FOR END INDEX 1177	1151	0662	4005		STC XPNT2		
1153 0664 0471 APO I / SKIP IF OTHER-TYPE EMPTY 1154 0665 6710 JMP DEL030 GG RE-WRITE INDEX 1156 0667 5757 5757 1157 0670 1044 STA XPNT / ZAP FIRST WORD OF NAME 1160 0671 1064 STA I XPNT / ZAP FIRST WORD OF NAME 1161 0672 1064 STA I XPNT / ZAP 1162 0673 1064 STA I XPNT / ZAP 1163 // WAS THAT THE LAST FILE // 1164 // WAS THAT THE LAST FILE // 1165 // WAS THAT THE LAST FILE // 1164 // WAS THAT THE LAST FILE // 1165 // WAS THAT THE LAST FILE // 1166 0674 0064 SET I XPNT / POINT TO START OF INDEX 1167 0676 1640 DEL010, SAE I XPNT / SET NON-EMPTY ENTRY 1170 0670 6676 JMP DEL030 / YES - REWRITE 100P 1172 0700 2044 XEX XPNT	1152	0663	1005		LDA XPNT2	/	PICK LENGTH OF OTHER-TYPE FILE
1154 0666 6710 JMP DEL030 / GO RE-WRITE INDEX 1155 0666 1020 LDA I / BOTH TYPES EMPTY CLEAR NAME 1156 0667 5757 5757 1157 0670 1044 STA XPNT / ZND 1160 0671 1064 STA I XPNT / ZND 1161 0672 1064 STA I XPNT / SRD 1162 0673 1064 STA I XPNT / SRD 1164 0671 1064 STA I XPNT / SRD 1166 0674 064 STA I XPNT / SRD 1166 / HAS THAT THE LAST FILE / 1166 / HAS THAT THE LAST FILE / 1166 / HAS THAT THE LAST FILE / 1167 0676 100EX-1 INDEX-1 1170 0676 1044 SET I XPNT / POINT TO START OF INDEX 1173 0701 DEL018 NOT YET - LOOP / 1174 / INDEX-1 POINT TO INDEX / 1175 /	1153	0664	0471		APO I	/	SKIP IF OTHER-TYPE EMPTY
1156 0664 1020 LOA I / BOTH TYPES EMPTY CLEAR NAME 1156 0667 5757 5757 1157 0670 1044 STA XPNT / ZAP FIRST WORD OF NAME 1160 0671 1064 STA I XPNT / ZAP 1161 0672 1064 STA I XPNT / SRD 1162 0673 1064 STA I XPNT / ARD 1163 / WAS THAT THE LAST FILE / 1164 / WAS THAT THE LAST FILE / 1165 / INDEX-1 / STA FILE 1166 0674 0064 SET I XPNT / IS THERE A NON-EMPTY ENTRY 1166 0674 0064 SET I XPNT / IS THERE A NON-EMPTY ENTRY 1170 0675 0777 INDEX-1 / IS THERE A NON-EMPTY ENTRY 1171 0677 6710 UPD DEL030 / YES - REWRITE LOOP 1172 0701 6676 JMP DEL010 NOT YET - LOOP ////////////////////////////////////	1154	0665	671Ø		JMP DELØ3Ø	/	GO RE-WRITE INDEX
1156 0667 5757 5757 5757 1157 0678 1044 STA XPNT / ZAP FIRST WORD OF NAME 1160 0671 1064 STA I XPNT / 2ND 1161 0672 1064 STA I XPNT / 2ND 1162 073 1064 STA I XPNT / 4TH 1163 / 1166 0674 0064 SET I XPNT / POINT TO START OF INDEX 1166 0676 0777 INDEX-1 1170 0675 0777 INDEX-1 1170 0676 1464 DEL010, SAE I XPNT / IS THERE A NON-EMPTY ENTRY 1171 0677 6710 JMP DEL030 / YES - REWRITE 1172 0700 0204 XSK XPNT / CHECK FOR END 1173 0701 6676 JMP DEL010 / NOT YET - LOOP 1173 0701 6676 JMP DEL010 / NOT YET - LOOP 1174 / 1175 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1176 / 1177 0702 0011 CLR 1280 0703 0064 SET I XPNT / POINT TO INDEX 1281 2785 1064 STA I XPNT / POINT TO INDEX 1282 2785 1064 STA I XPNT / POINT TO INDEX 1283 2786 0204 XSK XPNT / LAST ONE ? 1284 0787 6705 JMP2 / NO - LOOP 1285 / 1284 0787 6705 JMP2 / NO - LOOP 1285 // 1284 0787 6705 JMP RENET / RETURN JMP FOR R/K ROUTINE 1284 0787 6705 JMP RTRN2 1284 0787 6705 JMP RTRN2 1284 0787 6705 JMP RTRN2 1284 0787 6705 JMP RTRN2 1284 0787 6705 JMP RTRN2 1285 // 1285 // 1286 // RE-WRITE INDEX 1287 // 1286 // 1287 // 1286 // 1286 // 1287 // 1286 // 1288 // 1289 // 1280 // 1	1155	Ø666	1020		LDA I	/	BOTH TYPES EMPTY CLEAR NAME
1157 06/0 1044 STA XPNT / ZAP FIRST WORD OF NAME 1160 06/1 1064 STA I XPNT / ZND 1161 06/2 1064 STA I XPNT / SRD 1162 06/3 1064 STA I XPNT / SRD 1163 / WAS THAT THE LAST FILE / 1164 / WAS THAT THE LAST FILE / 1165 / NON-EMPTY ENTRY 10DEX 1166 06/7 0064 SET I XPNT / IS THERE A NON-EMPTY ENTRY 1167 06/7 0710 JMP DEL030 / YES - REWRITE 1172 0701 66/7 JMP DEL010 / NOT YET - LOOP 1173 0701 66/7 JMP DEL010 / NOT YET - LOOP 1174 / INDEX I S EMPTY: MAKE IT A NON-INDEX / 1175 / INDEX I S EMPTY: MAKE IT A NON-INDEX / 1174 / INDEX I S EMPTY: MAKE IT A NON-INDEX / 1175 / INDEX I S EMPTY: MAKE IT A NON-INDEX / 1176 / INDEX I S EMPTY: MAKE IT A NON-INDEX /	1156	0667	5757		5757		
1160 0671 1064 STA I XPNT / 2ND 1161 0672 1064 STA I XPNT / 4TH 1162 0673 1064 STA I XPNT / 4TH 1163 /	1157	0670	1044		STA XPNT	1	ZAP FIRST WORD OF NAME
1161 0672 1064 STA I XPN1 / 3RD 1162 0673 1064 STA I XPNT / 4TH 1163 / WAS THAT THE LAST FILE 1166 0674 0064 SET I XPNT / POINT TO START OF INDEX 1166 0674 0064 SET I XPNT / POINT TO START OF INDEX 1167 0676 0777 INDEX-1 1170 0676 1464 DELØ10, SAE I XPNT / IS THERE A NON-EMPTY ENTRY 1171 0677 6710 JMP DELØ30 / YES - REWRITE 1172 0700 0204 XSK XPNT / CHECK FOR END 1173 0701 6676 JMP DELØ10 / NOT YET - LOOP 1174 / / / / 1175 / INDEX IS EMPTY: MAKE IT A NON-INDEX / 1176 / / / / 1176 / / NOEX-1 / 1202 2704 3777 INDEX-1 / 1202 2706 0204 XSK XPNT / LAST ONE 2 1224	1160	06/1	1064		STA I XPNT	1	2ND
1162 06/3 1064 STA I XPNT / 4TH 1163 / 1164 / WAS THAT THE LAST FILE 1165 / 1166 0674 0064 SET I XPNT / POINT TO START OF INDEX 1167 0675 0777 INDEX-1 / IS THERE A NON-EMPTY ENTRY 1171 0676 1464 DEL010, SAE I XPNT / IS THERE A NON-EMPTY ENTRY 1172 0700 0204 XSK XPNT / CHECK FOR END 1173 0701 6676 JMP DEL010 / NOT YET - LOOP 1174 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1175 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1176 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1176 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1201 C704 3024 SET I XPNT / POINT TO INDEX 1202 2705 0064 SET I XPNT / POINT TO INDEX 1203 2706 0204 XSK XPNT / LAST ONE ? 1204 2707 6705 JMP -2 / NO - LOOP 12	1161	0672	1064		STA I XPNI	1	3RD
1163 / WAS THAT THE LAST FILE 1164 / WAS THAT THE LAST FILE 1165 / SET I XPNT / POINT TO START OF INDEX 1167 0675 0777 INDEX-1 1170 0676 1464 DELØ1Ø, SAE I XPNT / IS THERE A NON-EMPTY ENTRY 1171 0676 1464 DELØ1Ø, SAE I XPNT / IS THERE A NON-EMPTY ENTRY 1171 0676 1464 DELØ1Ø, SAE I XPNT / CHECK FOR END 1172 0700 0204 XSK XPNT / CHECK FOR END 1173 0701 6676 JMP DELØ1Ø / NOT YET - LOOP 1174 / INDEX IS EMPTY: MAKE IT A NON-INDEX / 1175 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1176 / INDEX-1 / 1200 2703 0064 SET I XPNT / POINT TO INDEX 1201 2704 3777 INDEX-1 / 1202 2705 0064 SET I XPNT / LAST ONE ? 1203 0706 0204 XSK XPNT / LAST ONE ? 1204 7707 </td <td>1162</td> <td>06/3</td> <td>1064</td> <td></td> <td>STA I XPNT</td> <td>/</td> <td>41H</td>	1162	06/3	1064		STA I XPNT	/	41H
1164 / WAS THAT THE LAST FILE 1165 / 1166 0674 0064 SET I XPNT / POINT TO START OF INDEX 1170 0675 0777 INDEX-1 1170 0676 1464 DEL010, SAE I XPNT / IS THERE A NON-EMPTY ENTRY 1171 0677 6710 JMP DEL030 / YES - REWRITE 1172 0700 0204 XSK XPNT / CHECK FOR END 1173 0701 6676 JMP DEL010 / NOT YET - LOOP 1174 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1175 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1176 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1176 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1201 2702 0011 CLR 1202 2705 1064 STA I XPNT / POINT TO INDEX 1203 2706 0204 XSK XPNT / LAST ONE ? 1204 0726 JMP2 / NO - LOOP 1205 / RE-wRITE INDEX 1206 / RE-wRITE INDEX <td>1163</td> <td></td> <td></td> <td></td> <td></td> <td>• •</td> <td></td>	1163					• •	
1100 0674 0064 SET 1 XPNT / POINT TO START OF INDEX 1167 0675 0777 INDEX-1 1170 0676 1464 DELØ1Ø, SAE 1 XPNT / IS THERE A NON-EMPTY ENTRY 1171 0676 1644 DELØ1Ø, SAE 1 XPNT / IS THERE A NON-EMPTY ENTRY 1171 0676 1644 DELØ1Ø, SAE 1 XPNT / CHECK FOR END 1172 0700 0204 XSK XPNT / CHECK FOR END 1173 0701 6676 JMP DELØ1Ø / NOT YET - LOOP 1174 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1175 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1176 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1176 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1201 3704 3777 INDEX-1 1202 2705 1064 STA I XPNT / ZERO ONE WORD 1203 2706 0264 XSK XPNT / LAST ONE ? 1204 0727 6705 JMP2 / NO - LOOP 1205 / / RE-wRITE INDE	1104			1	WAS THAT THE LA	151	FILE .
1160 0074 0064 SET 1 XPNT / POINT TO START OF INDEX 1167 0076 1464 DEL010, SAE I XPNT / IS THERE A NON-EMPTY ENTRY 1171 0076 1464 DEL010, SAE I XPNT / IS THERE A NON-EMPTY ENTRY 1172 0070 0204 XSK XPNT / CHECK FOR END 1173 0701 6676 JMP DEL010 / NOT YET - LOOP 1174 / INDEX.IS EMPTY: MAKE IT A NON-INDEX 1175 / INDEX.IS EMPTY: MAKE IT A NON-INDEX 1176 / INDEX.IS EMPTY: MAKE IT A NON-INDEX 1176 / INDEX.IS EMPTY: MAKE IT A NON-INDEX 1176 / INDEX.IS EMPTY: MAKE IT A NON-INDEX 1177 2702 0011 CLR 1200 2703 0064 SET I XPNT / POINT TO INDEX 1202 2705 1064 STA I XPNT / ZERO ONE WORD 1203 2706 0204 XSK XPNT / LAST ONE ? 1204 0727 6705 JMP2 / NO - LOOP 1205 / / RE-WRITE INDEX / <t< td=""><td>1144</td><td>0471</td><td>010141</td><td>/</td><td>CET I VONT</td><td>,</td><td>DOINT TO START OF INDEX</td></t<>	1144	0471	010141	/	CET I VONT	,	DOINT TO START OF INDEX
1170 0675 0777 INDEX-1 1170 0676 1464 DELØ1Ø, SÆE I XPNT / IS THERE A NON-EMPTY ENTRY 1171 0677 671Ø JMP DELØ3Ø / YES - REWRITE 1172 070Ø 0204 XSK XPNT / CHECK FOR END 1173 070Ø 6676 JMP DELØ1Ø / NOT YET - LOOP 1174 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1175 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1177 0702 0011 CLR 1200 2703 0664 SET I XPNT / POINT TO INDEX 1201 C704 3777 INDEX-1 ZERO ONE WORD 1202 2705 1064 SET I XPNT / LAST ONE ? 1203 2706 0204 XSK XPNT / LAST ONE ? 1204 0707 6705 JMP2 / NO - LOOP 1205 / RE-wRITE INDEX / / 1210 2710 0262 DELØ3Ø, SET I RDWR / SET WRITE CODE 1212 2712 2077 SET I RETURN / RETURN JUMP FOR R/K R	1167	0475			SET 1 XPNU	/	PUINT TU START OF INDEX
1171 0070 1044 DEL010 JMP DEL030 / YES - REWRITE 1172 0700 0204 XSK XPNT / CHECK FOR END 1173 0701 6676 JMP DEL010 / NOT YET - LOOP 1174 / / INDEX IS EMPTY: MAKE IT A NON-INDEX 1175 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1176 / 1177 0702 0011 CLR 1200 2703 0264 SET I XPNT / POINT TO INDEX 1201 2704 0777 INDEX-1 ////////////////////////////////////	1170	0676	1464	051010	INDEX-1	,	IS THERE & NON-EMPTY ENTRY
1172 0700 0204 XSK XPNT / CHECK FOR END 1173 0701 6676 JMP DEL010 / NOT YET - LOOP 1174 / / 1175 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1176 / 1177 0702 0011 1200 2703 0064 SET I XPNT / POINT TO INDEX 1201 2704 3777 INDEX-1 1202 2705 1064 STA I XPNT / ZERO ONE WORD 1203 2706 0204 XSK XPNT / LAST ONE ? 1204 3727 6705 JMP2 / NO - LOOP 1205 / RE-WRITE INDEX / 1206 / RE-WRITE INDEX / 1207 / 0 SET I RDWR / SET WRITE CODE 1210 2710 0262 DEL030, SET I RDWR / SET WRITE CODE 1211 2711 0704 724 / 1212 2712 2M77 SET I RETURN / RETURN JUMP FOR R/K ROUTINE 1213 2713 6543	1171	0677	6710	DECEID,	IMP DEL 030	1	YES - DEWRITE
1172 0701 6676 JMP DEL010 / NOT YET - LOOP 1174 / 1175 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1176 / 1 / 1176 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1177 0702 0011 CLR 1200 2703 0064 SET I XPNT / POINT TO INDEX 1201 2704 0777 INDEX-1 / 1202 2706 0204 XSK XPNT / LAST ONE ? 1204 0707 6705 JMP2 / NO - LOOP 1205 / / RE-wRITE INDEX / 1206 / RE-wRITE INDEX / SET WRITE CODE 1210 2710 0262 DEL030, SET I RDWR / SET WRITE CODE 1211 2711 0704 724 / RETURN / RETURN / RETURN JUMP FOR R/W ROUTINE 1212 2712 0077 SET I RETURN / RETURN JUMP FOR R/W ROUTINE / 1213 2713 0543 JMP RTR2 / / 1214 2714 026	1172	0700	0204		YSK YPNT	',	CHECK FOR END
1174 / 1174 / 1175 / 1176 / 1177 0702 1176 / 1200 2703 0204 SET I XPNT 1201 C104 0202 0703 0204 SET I XPNT 1201 C704 0202 0705 1203 0706 0204 XSK XPNT 1205 / 1206 / 1207 / 1208 2705 1209 2706 1204 XSK XPNT 1205 / 1206 / 1207 / 1210 2710 0262 1211 2711 0704 1212 2712 0077 1213 2713 0543 1214 2714 0261 1215 2715 0277 1216 2716 0244 1217 XPARM <td< td=""><td>1173</td><td>0701</td><td>6676</td><td></td><td></td><td>',</td><td>NOT YET - LOOP</td></td<>	1173	0701	6676			',	NOT YET - LOOP
1175 / INDEX IS EMPTY: MAKE IT A NON-INDEX 1176 / 1176 / 1177 0702 0011 CLR 1200 0703 0064 SET I XPNT / POINT TO INDEX 1201 0704 0777 INDEX-1 1202 2705 1064 STA I XPNT / ZERO ONE WORD 1203 0706 0204 XSK XPNT / LAST ONE ? 1204 0707 6705 JMP2 / NO - LOOP 1205 / / 1207 / 1210 0710 0262 DEL030, SET I RDWR / SET WRITE CODE 1211 2712 0077 SET I RETURN / RETURN JUMP FOR R/W ROUTINE 1212 2712 0077 SET I RETURN / RETURN JUMP FOR R/W ROUTINE 1213 2713 6543 JMP RTRN2 ////////////////////////////////////	1174	0101	0070	/	Sin DECEIE	,	
1176 / 1177 0702 0011 CLR 1200 0703 0064 SET I XPNT / POINT TO INDEX 1201 0704 0777 INDEX-1 1202 0705 1064 STA I XPNT / ZERO ONE WORD 1203 0706 0204 XSK XPNT / LAST ONE ? 1204 0707 6705 JMP2 / NO - LOOP 1205 /	1175			,	INDEX IS EMPTY:	MΔ	KE IT A NON-INDEX
1177 0702 0011 CLR 1200 0703 0064 SET I XPNT / POINT TO INDEX 1201 0704 0777 INDEX-1 1202 0705 1064 STA I XPNT / ZERO ONE WORD 1203 0706 0204 XSK XPNT / LAST ONE ? 1204 0707 6705 JMP2 / NO - LOOP 1205 /	1176				INDEX ID LOUT		
1200 2703 0064 SET I XPNT / POINT TO INDEX 1201 2704 0777 INDEX-1 1202 2705 1064 STA I XPNT / ZERO ONE WORD 1203 0706 0204 XSK XPNT / LAST ONE ? 1204 0707 6705 JMP2 / NO - LOOP 1205 / RE-wRITE INDEX / 1206 / RE-wRITE INDEX / 1210 2710 0262 DEL030, SET I RDWR / SET WRITE CODE 1211 2711 0704 704 1212 2712 0477 SET I RETURN / RETURN JUMP FOR R/W ROUTINE 1213 2713 6543 JMP RTRN2 / 1214 2714 0261 SET I PARAM / 1215 2715 0277 XPARM / 00 THE WRITE THING, AND RETURN 10 CALLES 1216 2716 6344 JMP RWENT / 00 THE WRITE THING, AND RETURN 10 CALLES	1177	0702	0011		CLR		
1201 2704 3777 INDEX-1 1202 2705 1064 STA I XPNT / ZERO ONE WORD 1203 0706 0204 XSK XPNT / LAST ONE ? 1204 0707 6705 JMP2 / NO - LOOP 1205 / / RE-wRITE INDEX / 1206 / RE-wRITE INDEX / 1210 0710 0262 DEL030, SET I RDWR / SET WRITE CODE 1211 0711 0704 704 / 1212 0712 0477 SET I RETURN / RETURN JUMP FOR R/W ROUTINE 1213 0713 6543 JMP RTRN2 / 1214 0/14 0/61 SET I PARAM / 1215 0/715 0/277 XPARM / D0 THE WRITE THING, AND RETURN TO CALLES 1216 0/16 6344 JMP RWENT / D0 THE WRITE THING, AND RETURN TO CALLES	1200	2703	0064		SET I XPNT	1	POINT TO INDEX
1202 2705 1064 STA I XPNT / ZERO ONE WORD 1203 2706 0204 XSK XPNT / LAST ONE ? 1204 0707 6705 JMP2 / NO - LOOP 1205 / 1206 / RE-WRITE INDEX 1207 / 1210 2710 0262 DEL030, SET I RDWR / SET WRITE CODE 1211 2711 0704 724 1212 2712 2077 SET I RETURN / RETURN JUMP FOR R/W ROUTINE 1213 2713 6543 JMP RTRN2 1214 2714 0261 SET I PARAM 1215 2715 0277 XPARM 1216 2716 6344 JMP RWENT / DO THE WRITE THING, AND RETURN TO CALLEE FIECT	1201	2704	0777		INDEX-1		
1203 0706 0204 XSK XPNT / LAST ONE ? 1204 0707 6705 JMP2 / NO - LOOP 1205 / 1206 / RE-WRITE INDEX 1207 / 1210 0710 0262 1211 0711 0704 1212 0712 04 1212 0712 04 1213 0713 6543 1214 0714 0261 1215 0715 0277 1216 0716 6344 1217 00 THE WRITE THING, AND RETURN TO CALLER	1202	27 05	1064		STA I XPNT	1	ZERO ONE WORD
1204 0707 6705 JMP2 / NO - LOOP 1205 / 1206 / RE-WRITE INDEX 1207 / 1210 2710 0262 DEL030, SET I RDWR / SET WRITE CODE 1211 2711 0704 704 1212 2712 0077 SET I RETURN / RETURN JUMP FOR R/W ROUTINE 1213 2713 6543 JMP RTRN2 1214 2714 0261 SET I PARAM 1215 2715 0277 XPARM 1216 2716 6344 JMP RWENT / DO THE WRITE THING, AND RETURN TO CALLEE EVENT	1203	2706	0204		XSK XPNT	1	LAST ONE ?
1205 / 1206 / 1207 / 1210 2710 0062 1211 2711 0704 1212 2712 0077 1213 2713 6543 1214 2714 0061 1215 2715 0277 1216 2716 6344 1217 2716 6344 1217 2716 6344 1217 2716 6344	1204	0707	6705		JMP2	1	NO - LOOP
1206 / RE-WRITE INDEX 1207 / 1210 2710 0062 DEL030, SET I RDWR / SET WRITE CODE 1211 2711 0704 704 1212 2712 0077 SET I RETURN / RETURN JUMP FOR R/W ROUTINE 1213 2713 6543 JMP RTRN2 1214 2714 0061 SET I PARAM 1215 2715 0277 XPARM 1216 2716 6344 JMP RWENT / D0 THE WRITE THING, AND RETURN TO CALLEE 1217 FURCT 200 THE WRITE THING, AND RETURN TO CALLEE 500 THE WRITE THING, AND RETURN TO CALLEE	1205			1			
1207 / 1210 2710 0062 DEL030, SET I RDWR / SET WRITE CODE 1211 2711 0704 704 1212 2712 0077 SET I RETURN / RETURN JUMP FOR R/W ROUTINE 1213 2713 6543 JMP RTRN2 1214 2714 0061 SET I PARAM 1215 2715 0277 XPARM 1216 2716 6344 JMP RWENT / D0 THE WRITE THING, AND RETURN TO CALLER	1206			/ R	E-WRITE INDEX		
1210 2710 0062 DEL030, SET I RDWR / SET WRITE CODE 1211 2711 0704 704 1212 2712 0077 SET I RETURN / RETURN JUMP FOR R/W ROUTINE 1213 2713 6543 JMP RTRN2 1214 2714 0061 SET I PARAM 1215 2715 0277 XPARM 1216 2716 6344 JMP RWENT / D0 THE WRITE THING, AND RETURN TO CALLER	1207			1			
1211 2711 0704 704 1212 2712 0077 SET I RETURN / RETURN JUMP FOR R/W ROUTINE 1213 2713 6543 JMP RTRN2 1214 2714 0061 SET I PARAM 1215 2715 0277 XPARM 1216 2716 6344 JMP RWENT / DO THE WRITE THING, AND RETURN TO CALLER FUECT	1210	2710	0062	DEL030,	SET I RDWR	1	SET WRITE CODE
1212 2712 2077 SET I RETURN / RETURN JUMP FOR R/W ROUTINE 1213 2713 6543 JMP RTRN2 1214 2714 2061 SET I PARAM 1215 2715 0277 XPARM 1216 2716 6344 JMP RWENT / DO THE WRITE THING, AND RETURN TO CALLER	1211	2711	0704		784		
1213 2713 6543 JMP RTRN2 1214 2714 8061 SET I PARAM 1215 8715 8277 XPARM 1216 8716 6344 JMP RWENT / DO THE WRITE THING, AND RETURN TO CALLER 1217	1212	2712	2077		SET I REJURN	1	RETURN JUMP FOR RZW ROUTINE
1214 0714 0761 SET I PARAM 1215 0715 0277 XPARM 1216 0716 6344 JMP RWENT / DO THE WRITE THING, AND RETURN TO CALLER 1217 EVECT	1213	2713	6543		JMP RTRN2		
1215 3715 M277 XPARM 1216 3716 6344 JMP RWENT / DO THE WRITE THING, AND RETURN TO CALLER 1217	1214	2714	0261		SET I PARAM		
1216 2716 6344 JMP RWENT / DO THE WRITE THING, AND RETURN TO CALLER	1215	27.15	V1277		XPARM		
		2716	6344		JMP RWENT	1	DO THE WRITE THING, AND RETURN TO CALLER
	1216	1. A.			-		

1221		/	THIS WILL CAUSE AN ASSEMBLY ERROR IF ROUTINE BECOMES SU LARGE
1222		1	THAT INDEX WILL OVERLAY CODE WHEN IT IS READ IN
1223	· · · · · · · · · · · · · · · · · · ·		ASMIFM INDEX
1224			NAUGHTY BAD BOY - ROUTINE IS TOO BIG FOR ONE FIELD.
1225		1	IN CASE OF MINOR SIZE PROBLEMS, REMOVE 5 LINES OF CODE AT FORMAL.
1226		/	REMOVE 3 LINES AT FSP000, AND THREE LINES AT FSP035. THIS SHOWED
1227		1	REMOVE ALL REFERENCES TO FOF050. ALL OTHER TAGS MUST BE RETAINED.
1230		1	
1231		1	
1232		1	
1233		1	END OF FRED
1234		1	

P.14



DEL000 4641 DEL010 4676 DEL030 4710 ENTSW 0011 ERRTN 4252 EXT 4254 FCF000 4550 FCF010 4554 FCF030 4576 FCF040 4602 FCF050 4605 FCF060 4630 FCF070 4633 FDV 0002 FSP000 4345 FSP010 4370 FSP020 4404 FSP035 4425 FSP040 4432 FSP040 4432 FSP050 4444 FSP050 4444 FSP060 451 FSP070 4555 FSP080 4510 FSP080 4510 FSP070 4525 FSP100 4525 FSP10 4542 GETFLD 4116 GTF010 4140 INDEX 1000 IOBLK 4100 IOLOOP 4071 LKP000 4144 LKP010 4174 LKP020 4215 LKP030 4215 LKP030 4215 LKP030 4216 LRIF 6224 LRMF 6244 MARK 0210 MEMADD 4072		
DEL010 4676 DEL030 4710 ENTSW 0011 ERRTN 4252 EXT 4254 FCF000 4552 FCF010 4554 FCF030 4576 FCF030 4602 FCF050 4605 FCF050 4630 FCF070 4633 FDV 0002 FSP000 4345 FSP010 4370 FSP020 4404 FSP035 4425 FSP040 4432 FSP040 4432 FSP050 4444 FSP050 4444 FSP060 451 FSP070 4555 FSP080 4510 FSP080 4510 FSP080 4510 FSP080 4525 FSP100 4525 FSP100 4525 FSP10 4542 GETFLD 4116 GTF010 4140 INDEX 1000 IOBLK 4100 IOLOOP 4071 LKP000 4144 LKP010 4174 LKP030 4215 LKP030 4215 LKP030 4215 LKP030 4216 LKP030 4216 LKP030 4212 LKP030 4212 LKP1 0206 LP2 0207 LRIB 6224 LRMF 6224 LRMF 6224 LRMF 6224		16
DEL030 4710 ENTSW 0011 ERRTN 4252 EXT 4054 FCF000 4550 FCF010 4554 FCF030 4576 FCF040 4602 FCF050 4605 FCF070 4633 FDV 0002 FSP000 4345 FSP010 4370 FSP020 4404 FSP030 4413 FSP035 4425 FSP040 4432 FSP050 4444 FSP060 4451 FSP070 4525 FSP080 4510 FSP080 4510 FSP090 4521 FSP100 4525 FSP100 4525 FSP100 4525 FSP100 4525 FSP100 4542 GETFLD 4116 GTF010 4140 INDEX 1000 IOBLK 4100 IOLOOP 4071 LKP020 4213 LKP030 4215 LKP030 4215 LKP030 4215 LKP030 4215 LKP050 4236 LOOKUP 4142 LP1 0006 LP2 0007 LRIB 6234 LRIF 6224 LRMF 6244 MARK 0010		
ENTSW 0011 ERRTN 4252 EXT 4054 FCF000 4550 FCF010 4554 FCF030 4576 FCF040 4602 FCF050 4605 FCF070 4633 FDV 0002 FSP000 4345 FSP010 4370 FSP020 4404 FSP030 4413 FSP035 4425 FSP035 4425 FSP040 4432 FSP050 4444 FSP060 4451 FSP070 4455 FSP040 4525 FSP100 4525 FSP100 4525 FSP100 4525 FSP100 4525 FSP100 4525 FSP100 4542 GETFLD 4116 GTF010 4140 INDEX 1000 IOBLK 4100 IOLOOP 4071 LKP000 4144 LKP010 4174 LKP020 4213 LKP030 4215 LKP050 4236 LOOKUP 4142 LP1 0006 LP2 0007 LRIB 6234 LRIF 6224 LRMF 6244 MARK 0010		
ERRTN 4252 EXT 4254 FCF000 4550 FCF010 4554 FCF030 4576 FCF040 4602 FCF050 4605 FCF060 4630 FCF070 4633 FDV 0002 FSP000 4345 FSP010 4370 FSP020 4404 FSP030 4413 FSP030 4413 FSP035 4425 FSP040 4432 FSP040 4432 FSP060 4451 FSP060 4451 FSP060 4451 FSP060 4510 FSP090 4525 FSP100 4525 F		
EXT 4254 FCF000 4552 FCF010 4554 FCF030 4576 FCF040 4602 FCF050 4605 FCF070 4633 FDV 0002 FSP000 4345 FSP010 4370 FSP020 4404 FSP030 4413 FSP030 4413 FSP035 4425 FSP040 4432 FSP050 4444 FSP060 4451 FSP060 4451 FSP070 4555 FSP080 4510 FSP070 4525 FSP10 4525 FSP110 4542 GETFLD 4116 GTF010 4140 INDEX 1000 IOBLK 4100 IOBLK 4100 IOBLK 4100 IOBLK 4100 IOBLK 4100 IOBLK 4100 IOBLK 4100 IOBLK 412 KP030 4215 KP030		
CF000 4552 CF010 4554 CF030 4576 CF050 4602 CF050 4605 CF070 4633 DV 0002 SP000 4345 SP010 4370 SP020 4404 SP035 4425 SP030 4413 SP040 4432 SP050 4444 SP060 455 SP080 4510 SP080 4510 SP090 4521 SP100 4525 SP110 4542 GETFLD 4116 STP00 4525 SP110 4542 GETFLD 4116 STP00 4215 KP000 4144 KP010 4174 KP020 4213 KP030 4215 KP030 4215 KP050 4236 OOKUP 4142 P1 0206 P2 0207		
CF010 4554 CF030 4576 CF040 4602 CF050 4605 CF070 4633 DV 0002 SP000 4345 SP010 4370 SP020 4404 SP030 4413 SP035 4425 SP040 4432 SP050 4444 SP060 4451 SP060 4510 SP070 4555 SP100 4525 SP110 4525 SP110 4542 SETFLD 4116 STF010 4140 NDEX 1000 OBLK 4100 ODLX 1000 OBLK 4100 ODLX 1000 A144 KP010 4174 KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6224 RMF 6244 ARK 0010		
CF030 4576 CF040 4602 CF050 4605 CF060 4630 CF070 4633 DV 0002 SP000 4345 SP010 4370 SP020 4404 SP030 4413 SP035 4425 SP040 4451 SP050 4444 SP060 4451 SP070 4455 SP080 4510 SP090 4521 SP100 4525 SP110 4542 SFTFLD 4116 STF010 4140 NDEX 1000 OBLK 4100 OBLK 4100 OLOOP 4071 KP000 4144 KP010 4174 KP020 4213 KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6224 RMF 6244 ARK 0010		
CF040 4602 CF050 4605 CF070 4633 DV 0002 SP000 4345 SP010 4370 SP020 4404 SP030 4413 SP035 4425 SP040 4432 SP050 4444 SP060 4451 SP070 4455 SP080 4510 SP090 4521 SP100 4525 SP110 4542 SFTFLD 4116 STF010 4140 INDEX 1000 OBLK 4100 OLOOP 4071 KP000 4144 KP010 4174 KP020 4213 KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6224 RMF 6244 ARK 0010		
CF050 4605 CF070 4633 DV 0002 SP000 4345 SP010 4370 SP020 4404 SP030 4413 SP035 4425 SP040 4432 SP050 4444 SP060 4451 SP070 4455 SP080 4510 SP090 4521 SP100 4525 SP110 4542 SFTFLD 4116 STF010 4140 NDEX 1000 OBLK 4100 OLOOP 4071 KP000 4144 KP010 4174 KP020 4213 KP030 4215 KP030 4216 KP030 4216 KP030 4216 KP030 4272		
CF060 4630 CF070 4633 DV 0002 SP000 4345 SP010 4370 SP020 4404 SP030 4413 SP035 4425 SP040 4432 SP050 4441 SP060 4451 SP070 4455 SP080 4510 SP090 4521 SP100 4525 SP110 4542 ETFLD 4116 STF010 4140 NDEX 1000 OBLK 4100 OLOP 4071 KP000 4144 KP010 4174 KP020 4213 KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6224 RMF 6244 ARK 0010		
CF070 4633 DV 0002 SP000 4345 SP010 4370 SP020 4404 SP030 4413 SP035 4425 SP040 4432 SP050 4444 SP060 4451 SP070 4455 SP080 4510 SP090 4521 SP100 4525 SP110 4542 ETFLD 4116 TF010 4140 NDEX 1000 OBLK 4100 OLOOP 4071 KP000 4144 KP010 4174 KP020 4213 KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6224 RMF 6244 ARK 0010 EMADD 4072		
DV 0002 SP000 4345 SP010 4370 SP020 4404 SP030 4413 SP035 4425 SP040 4432 SP050 4444 SP060 4451 SP070 4455 SP080 4510 SP090 4521 SP100 4525 SP110 4542 ETFLD 4116 TF010 4140 NDEX 1000 OBLK 4100 OLOOP 4071 KP000 4144 KP010 4174 KP020 4213 KP030 4215 KP050 4236 OOKUP 4142 P1 0206 P2 0207 RIB 6234 RIF 6224 RMF 6244 ARK 0010 EMADD 4072		
SPØØØ 4345 SPØ10 4370 SPØ20 4404 SPØ30 4413 SPØ35 4425 SPØ40 4432 SPØ50 4444 SPØ60 4451 SPØ60 4451 SPØ60 4451 SPØ70 4455 SPØ90 4521 SP100 4525 SP110 4542 ETFLD 4116 TFØ10 4140 NDEX 1000 OBLK 4100 OLOOP 4071 KPØ00 4144 KPØ10 4174 KPØ20 4213 KPØ50 4236 OOKUP 4142 P1 0206 P2 0207 RIB 6234 RIF 6244 ARK 0210 EMADD 4272		
SP010 4370 SP020 4404 SP030 4413 SP035 4425 SP040 4432 SP050 4444 SP060 4451 SP060 4451 SP070 4455 SP080 4510 SP090 4521 SP100 4525 SP110 4542 ETFLD 4116 TF010 4140 NDEX 1000 OLOOP 4071 KP000 4144 KP010 4174 KP020 4213 KP030 4215 KP030 4215 KP050 4236 OOKUP 4142 P1 0206 P2 0207 R1B 6234 R1F 6224 ARK 0210 EMADD 4272		
SP020 4404 SP030 4413 SP035 4425 SP040 4432 SP050 4444 SP060 4451 SP060 4451 SP070 4455 SP080 4510 SP090 4521 SP100 4525 SP110 4542 ETFLD 4116 TFØ10 4140 NDEX 1000 OBLK 4100 OLOOP 4071 KP000 4144 KP010 4174 KP020 4213 KP030 4215 KP030 4215 KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6244 ARK 0010 EMADD 4072		
SP030 4413 SP035 4425 SP040 4432 SP050 4444 SP050 4444 SP060 4451 SP070 4455 SP080 4510 SP090 4521 SP100 4525 SP100 4525 SP110 4542 ETFLD 4116 TF010 4140 NDEX 1000 OBLK 4100 OLOOP 4071 KP000 4144 KP010 4174 KP020 4213 KP030 4215 KP030 4215 KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6244 ARK 0010 EMADD 4072		
SP035 4425 SP040 4432 SP050 4444 SP060 4451 SP060 4451 SP070 4455 SP080 4510 SP090 4521 SP100 4525 SP110 4542 ETFLD 4116 TF010 4140 NDEX 1000 OBLK 4100 OLOOP 4071 KP000 4144 KP010 4174 KP030 4215 KP030 4215 KP030 4215 KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6244 ARK 0010 EMADD 4072		
SP040 4432 SP050 4444 SP060 4451 SP070 4455 SP080 4510 SP090 4521 SP100 4525 SP110 4542 ETFLD 4116 TFØ10 4140 NDEX 1000 OBLK 4100 OLOOP 4071 KPØ00 4144 KP010 4174 KP020 4213 KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6244 ARK 0010 EMADD 4072		
SP050 4444 SP060 4451 SP070 4455 SP080 4510 SP090 4521 SP100 4525 SP110 4542 ETFLD 4116 TF010 4140 NDEX 1000 OBLK 4100 OLOOP 4071 KP000 4144 KP010 4174 KP020 4213 KP030 4215 KP050 4236 OOKUP 4142 P1 0206 P2 0207 RIB 6234 RIF 6244 ARK 0212 EMADD 4072		
SP060 4451 SP070 4455 SP080 4510 SP090 4521 SP100 4525 SP110 4542 ETFLD 4116 TF010 4140 NDEX 1000 OBLK 4100 OLOOP 4071 KP000 4144 KP010 4174 KP020 4213 KP020 4215 KP030 4215 KP030 4215 KP030 4215 KP030 4215 RP050 4236 OOKUP 4142 P1 0206 P2 0207 RIB 6234 RIF 6244 ARK 0012 EMADD 4072		
SP070 4455 SP080 4510 SP090 4521 SP100 4525 SP110 4542 ETFLD 4116 TF010 4140 NDEX 1000 OBLK 4100 OLOOP 4071 KP000 4144 KP010 4174 KP020 4213 KP030 4215 KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6224 RMF 6244 ARK 0010 EMADD 4072		
SP080 4510 SP090 4521 SP100 4525 SP110 4542 ETFLD 4116 TF010 4140 NDEX 1000 OBLK 4100 OLOOP 4071 KP000 4144 KP010 4174 KP020 4213 KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6244 ARK 0012 EMADD 4072		
SP090 4521 SP100 4525 SP110 4542 ETFLD 4116 TF010 4140 NDEX 1000 OBLK 4100 OBLK 4100 OLOOP 4071 KP000 4144 KP010 4174 KP020 4213 KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6224 RMF 6244 ARK 0010 EMADD 4072		
SP100 4525 SP110 4542 ETFLD 4116 TF010 4140 NDEX 1000 OBLK 4100 OLOOP 4071 KP000 4144 KP010 4174 KP020 4213 KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6224 RMF 6244 ARK 0010 EMADD 4072		
SP110 4542 ETFLD 4116 TF010 4140 NDEX 1000 OBLK 4100 OLOOP 4071 KP000 4144 KP010 4174 KP020 4213 KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6224 RMF 6244 ARK 0012 EMADD 4072		
BTFLD 4116 TFØ10 4140 NDEX 1000 OBLK 4100 OLOOP 4071 KP000 4144 KP010 4174 KP020 4213 KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6244 ARK 0012 EMADD 4072		
TF010 4140 NDEX 1000 OBLK 4100 OLOOP 4071 KP000 4144 KP010 4174 KP020 4213 KP030 4215 KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6224 RMF 6244 ARK 0010 EMADD 4072		
NDEX 1000 OBLK 4100 OLOOP 4071 KP000 4144 KP010 4174 KP020 4213 KP030 4215 KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6224 RMF 6244 ARK 0010 EMADD 4072		
OBLK 4100 OLOOP 4071 KP000 4144 KP010 4174 KP020 4213 KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6224 RMF 6244 ARK 0010 EMADD 4072		
OLOOP 4071 KP000 4144 KP010 4174 KP020 4213 KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6224 RMF 6244 ARK 0010 EMADD 4072		
KP000 4144 KP020 4213 KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6244 ARK 0010 EMADD 4072		
KPØ10 4174 KPØ20 4213 KPØ30 4215 KPØ50 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6244 ARK 0010 EMADD 4072		
KP020 4213 KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6224 RMF 6244 ARK 0010 EMADD 4072		
KP030 4215 KP050 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6224 RMF 6244 ARK 0010 EMADD 4072		
KPØ50 4236 OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6224 RMF 6244 ARK 0010 EMADD 4072		
OOKUP 4142 P1 0006 P2 0007 RIB 6234 RIF 6224 RMF 6244 ARK 0010 EMADD 4072		
P1 0006 P2 0007 RIB 6234 RIF 6224 RMF 6244 ARK 0010 EMADD 4072		
P2 0007 RIB 6234 RIF 6224 RMF 6244 ARK 0010 EMADD 4072		
RIB 6234 RIF 6224 RMF 6244 ARK ØØ10 EMADD 4072		
RIF 6224 RMF 6244 ARK ØØ10 EMADD 4072		
RMF 6244 ARK ØØ10 EMADD 4072		
ARK ØØ10 EMADD 4072		
EMADD 4072		
P/6		
4. A. Sama and A. Sa And A. Sama and A. Sam And A. Sama and A.		

 11 AT	10.11.6	
NF JM	0212	
NTRØG	321	
NTRØ10	4335	
NTRØ	4342	
NTRØ	4344	
PARAM	2001	
PARM2	0223	
RDWR	9565	4
READ	4232	
RETURN	2017	
RET2	0216	
RET3	0015	
RPLØØØ	4303	
RPL010	4306	
RTRNØ	4543	
RWENT	4044	•
SVTRY	4637	
TRY	4635	
TRYLEN	4636	
UPLIM	4642	
WNTB	4261	
WNTS	4263	
WORD1	4226	
WRITE	4035	
XPARM	4277	
XPNT	0004	
XPNT2	0005	

P.17