

FlexTM

Utilities

COPYRIGHT © 1978 BY
Technical Systems Consultants, Inc.
P.O. Box 2574
West Lafayette, Indiana 47906
All Rights Reserved

I. Volume 1

FIND
WORDS
TYPOS
SPLIT
LOW-UP
UP-LOW

IV. Volume 4

FILES
PRUL
DATE
RPT
ECHO
HECHO

II. Volume 2

DUMP
OLOAD
CHECK
CMPMEM
FILTYP
DUP

V. Volume 5

FLIST
PDEL
SLEEP
REMSPC
CONCAT
CONTIN

III. Volume 3

MAP
DIR
INSTALL
FREE
REPLACE
TEST

VI. Volume 6

INTEG
RECOVER
MEMTEST
MEMDUMP
MEMOVE
MEMFILL

FIND

The FIND command is used for finding all lines in a text file containing a specified string. It is faster to use FIND than to enter the editor to find strings.

DESCRIPTION

The general syntax of the FIND command is:

```
FIND,<file spec>,<string>
```

The file spec defaults to a TXT extension and to the working drive. The string may be any printable characters (non-control characters) and is terminated by the carriage return or end of line character. Upon execution, all lines containing the specified string will be printed on the terminal preceded by that lines line number. When finished, the total number of lines found containing the string will be printed. Following are a few examples.

```
+++FIND,TEXT,THIS IS A TEST  
+++FIND,BOOK.TXT,OHIO
```

The first example would find and display all lines in the file TEXT.TXT which contained the character string "THIS IS A TEST". The second example would search the file BOOK.TXT for the string "OHIO" and list all lines found.

```

*  

* FIND STRING UTILITY  

*  

*  

* COPYRIGHT (C) 1978 BY  

*  

* TECHNICAL SYSTEMS CONSULTANTS, INC.  

* P. O. BOX 2574  

* WEST LAFAYETTE, INDIANA 47906  

*  

*  

* GLOBAL VARIABLES

```

7082	EOL	EQU	\$7082	\$/AC02
7740	FCB	EQU	\$7740	\$/AD40
7803	FMSCLS	EQU	\$7803	\$/B403
7806	FMS	EQU	\$7806	\$/B406
7103	WARMs	EQU	\$7103	\$/AD03
710F	GETCHR	EQU	\$710F	\$/AD15
7112	PUTCHR	EQU	\$7112	\$/AD18
7118	PSTRNG	EQU	\$7118	\$/AD1E
711E	PCRLF	EQU	\$711E	\$/AD24
7121	NXTCH	EQU	\$7121	\$/AD27
7127	GETFIL	EQU	\$7127	\$/AD20
712D	SETEXT	EQU	\$712D	\$/AD33
7130	ADDBX	EQU	\$7130	\$/AD36
7133	OUTDEC	EQU	\$7133	\$/AD39
7139	OUTHEX	EQU	\$7139	\$/AD3C
713C	RPTERR	EQU	\$713C	\$/AD3F
0100		ORG	\$0100	\$/A100

0100 20 10 FIND BRA FIND0

* TEMPORARY STORAGE

0102 01	VN	FCB	1	VERSION NUMBER
0103 00 00	LINE	FDB	0	
0105 00 00	POINT	FDB	0	
0107 00	ZERO	FCB	0	
0108 00 00	LP	FDB	0	
010A 00 00	ENDP	FDB	0	
010C 00 00	STP	FDB	0	
010E 00 00	VALUE	FDB	0	
0110 00 00	COUNT	FDB	0	
0112 CE 02 76	FIND0	LDX	#11UF	SET POINTERS
0115 FF 01 05		STX	POINT	
0118 CE 02 F6		LDX	#1 B	
011B FF 01 08		STX	LI	
011E BD 71 1E		JSR	PI RLF	OUTPUT CR & LF
0121 CE 77 40		LDX	#1 CB	POINT TO FCB

FIND STRING UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 2

0124 BD 71 27	JSR	GETFIL	GET FILE NAME
0127 25 6F	BCS	ERROR6	ERROR?
0129 86 01	LDA A	#1	SET FOR READ
012B A7 00	STA A	0,X	
012D BD 71 2D	JSR	SETEXT	SET TXT EXTENSION
0130 CE 77 40	LDX	#FCB	
0133 BD 78 06	JSR	FMS	OPEN FILE
0136 26 41	BNE	FIND3	ERRORS?
0138 CE 02 76	LDX	#BUF	POINT TO STRING BUFFER
013B BD 71 21 FIND1	JSR	NXTCH	GET CHARACTER
013E 81 0D	CMP A	#\$D	IS IT TERM?
0140 27 0A	BEQ	FIND15	
0142 B1 70 82	CMP A	EOL	
0145 27 05	BEQ	FIND15	
0147 A7 00	STA A	0,X	PUT CHARACTER IN BUF
0149 08	INX		BUMP POINTER
014A 20 EF	BRA	FIND1	REPEAT TIL DONE
014C FF 01 0A FIND15	STX	ENDP	SAVE END POINTER
014F CE 77 40	LDX	#FCB	POINT TO FCB
0152 BD 78 06 FIND2	JSR	FMS	GET CHARACTER
0155 26 22	BNE	FIND3	ERRORS?
0157 FE 01 08	LDX	LP	GET POINTER
015A A7 00	STA A	0,X	SAVE CHARACTER
015C 08	INX		BUMP TO NEXT
015D FF 01 08	STX	LP	SAVE POSITION
0160 CE 77 40	LDX	#FCB	RESTORE TO FCB
0163 81 0D	CMP A	#\$D	END OF LINE?
0165 26 EB	BNE	FIND2	
0167 FE 01 03	LDX	LINE	POINT TO LINE
0168 08	INX		
016B FF 01 03	STX	LINE	INC LINE NUMBER
016E CE 02 F6	LDX	#LB	POINT TO LINE
0171 FF 01 08	STX	LP	SAVE POINTER
0174 FF 01 0C	STX	STP	
0177 20 41	BRA	FIND4	
0179 A6 01 FIND3	LDA A	1,X	GET ERROR NUMBER
017B 81 08	CMP A	#8	IS IT EOF?
017D 27 25	BEQ	FIND32	
017F A6 01 ERROR	LDA A	1,X	GET ERROR TYPE
0181 81 03	CMP A	#3	FILE EXISTS?
0183 27 09	BEQ	ERROR2	
0185 81 04	CMP A	#4	NO FILE?
0187 27 0A	BEQ	ERROR4	
0189 BD 71 3C	JSR	RPTERR	REPORT ERROR
018C 20 10	BRA	ERROR8	
018E CE 02 6A ERROR2	LDX	#FEST	POINT TO STRING
0191 20 08	BRA	ERROR7	
0193 CE 02 4B ERROR4	LDX	#MSST	POINT TO STRING
0196 20 03	BRA	ERROR7	
0198 CE 02 58 ERROR6	LDX	#ILST	POINT TO STRING
019B BD 71 18 ERROR7	JSR	PTRNG	PRINT STRING
019E BD 78 03 ERROR8	JSR	FMSCLS	CLOSE ALL
01A1 7E 71 03	JMP	WRMS	RETURN TO FLEX
01A4 BD 71 1E FIND32	JSR	PRLF	PRINT CR & LF

FIND STRING UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 3

01A7 CE 02 30	LDX #STR	POINT TO STRING
01AA BD 71 18	JSR PSTRU	
01AD CE 01 10	LDX #COUNT	POINT TO TOTAL
01B0 5F	CLR B	CLEAR FLAG
01B1 BD 71 33	JSR OUTDEC	OUTPUT TOTAL
01B4 BD 71 1E	JSR PCRLF	PRINT CR & LF
01B7 7E 71 03	FIND35 JMP	RETURN TO FLEX
01BA CE 02 76	FIND4 LDX	POINT TO STRING
01BD FF 01 05	STX POINT	
01C0 BC 01 0A	FIND5 CPX	END OF STRING?
01C3 27 38	BEQ FIND7	IF SO, MATCH!
01C5 A6 00	LDA A 0,X	GET CHARACTER
01C7 08	INX	BUMP TO NEXT
01C8 FF 01 05	STX POINT	SAVE POINTER
01CB FE 01 08	LDX LP	
01CE A1 00	CMP R 0,X	COMPARE CHARACTER
01D0 26 09	BNE FIND6	
01D2 08	INX	BUMP TO NEXT
01D3 FF 01 08	STX LP	SAVE POSITION
01D6 FE 01 05	LDX POINT	
01D9 20 E5	BRA FIND5	REPEAT PROCESS
01DB FE 01 0C	FIND6 LDX STP	POINT TO LINE
01DE A6 00	LDA A 0,X	GET CHARACTER
01E0 81 00	CMP A #\$D	IS IT CR?
01E2 27 0D	BEQ FIND65	
01E4 68	INX	BUMP TO NEXT
01E5 FF 01 0C	STX STP	SAVE POINTER
01E8 FF 01 08	STX LP	
01EB A6 00	LDA A 0,X	GET CHARACTER
01ED 81 00	CMP R #\$D	IS IT CR?
01EF 26 C9	BNE FIND4	
01F1 CE 02 F6	FIND65 LDX #LB	RESTORE POINTER
01F4 FF 01 08	STX LP	
01F7 CE 77 40	LDX #FCB	POINT TO FCB
01FA 7E 01 52	FIND2 JMP	REPEAT
01FD BD 71 1E	FIND7 JSR PCRLF	OUTPUT CR & LF
0200 CE 01 03	LDX #LINE	POINT TO LINE #
0203 C6 01	LDA B #1	SET FLAG
0205 BD 71 33	JSR OUTDEC	OUTPUT NUMBER
0208 86 2E	LDA A #'.'	OUTPUT '.'
020A BD 71 12	JSR PUTCHR	
020D CE 01 07	LDX #ZERO	OUTPUT ZEROES
0210 BD 71 39	JSR OUTHEX	
0213 86 3D	LDA A #'='	OUTPUT '='
0215 BD 71 12	JSR PUTCHR	
0218 CE 02 F6	LDX #LB	POINT TO LINE
021B A6 00	FIND8 LDA A 0,X	GET CHARACTER
021D 81 00	CMP A #\$D	END OF LINE?
021F 27 06	BEQ FIND9	
0221 BD 71 12	JSR PUTCHR	OUTPUT CHARACTER
0224 08	INX	BUMP TO NEXT
0225 20 F4	BRA FIND8	
0227 FE 01 10	FIND9 LDX COUNT	BUMP FIND COUNT
022A 08	INX	

FIND STRING UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 4

022B FF 01 10	STX	COUNT
022E 20 C1	BRA	FIND65 REPEAT

* STRINGS

0230 54	STR	FCC	'TOTAL STRING OCCURENCE IS '
024A 04		FCB	4
024B 4E	NSST	FCC	'NO SUCH FILE'
0257 04		FCB	4
0258 49	ILST	FCC	'ILLEGAL FILE NAME'
0269 04		FCB	4
026A 46	FEST	FCC	'FILE EXISTS'
0275 04		FCB	4

* BUFFERS

0276	BUF	RMB	128	256	?
02F6		LB	128	256	?

END FIND

NO ERROR(S) DETECTED

SYMBOL TABLE:

ADDBX	7130	BUF	0276	COUNT	0110	ENDP	010A	EOL	' 7082
ERROR	017F	ERROR2	018E	ERROR4	0193	ERROR6	0198	ERROR7	019B
ERROR8	019E	FCB	7740	FEST	026A	FIND	0100	FIND0	0112
FIND1	0138	FIND15	014C	FIND2	0152	FIND3	0179	FIND32	01A4
FIND35	01B7	FIND4	018A	FIND5	01C0	FIND6	01DB	FIND65	01F1
FIND7	01FD	FIND8	021B	FIND9	0227	FMS	7806	FMSCLS	7803
GETCHR	710F	GETFIL	7127	ILST	0258	LB	02F6	LINE	0103
LP	0108	NSST	024B	NXTCH	7121	OUTDEC	7133	OUTHEX	7139
PCRLF	711E	POINT	0105	PSTRNG	7118	PUTCHR	7112	RPTERR	713C
SETEXT	712D	STP	010C	STR	0230	VALUE	010E	VN	0102
WARM3	7103	ZERO	0107						

WORDS

The WORDS utility is used to get a total word and line count of a text file. It is very useful in document and report preparation in keeping track of the size of the file.

DESCRIPTION

The general syntax of the WORDS command is:

```
WORDS,<file spec>
```

where the file spec defaults to a TXT extension and to the working drive. Upon execution, WORDS will read the file specified, count all words and lines, and then report the totals to the terminal. A word is considered to be any group of characters separated by either spaces or carriage returns. An example will demonstrate the use of WORDS.

```
+++WORDS,CHAPTER1
```

This command line would cause all of the words of the file named CHAPTER1.TXT on the working drive to be counted and the totals displayed on the terminal.

```

* COUNT WORDS IN FILE UTILITY
*
*
* COPYRIGHT (C) 1979 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*

```

* GLOBAL VARIABLES

7740	FCB	EQU	\$7740	\$A840
7803	FMSCLS	EQU	\$7803	\$B403
7806	FMS	EQU	\$7806	\$B406
7103	WARMS	EQU	\$7103	\$A003
710F	GETCHR	EQU	\$710F	\$AD15
7112	PUTCHR	EQU	\$7112	\$A010
7118	PSTRNG	EQU	\$7118	\$A01E
711E	PCRLF	EQU	\$711E	\$AD24
7127	GETFIL	EQU	\$7127	\$AD2D
712D	SETEXT	EQU	\$712D	\$A033
7133	OUTDEC	EQU	\$7133	\$A069
713C	RPTERR	EQU	\$713C	\$AD3F
7600		ORG	\$7600	\$A100

* PROGRAM STARTS HERE

7600 20 05	WORD0	BRA	WORD0
------------	-------	-----	-------

* TEMP STORAGE

7602 01	VN	FCB	1	
7603 00 00	COUNT	FDB	0	
7605 00 00	LCOUNT	FDB	0	
7607 CE 00 01	WORD0	LDX	#1	INIT COUNT
760A FF 76 03		STX	COUNT	
760D CE 77 40		LDX	#FCB	POINT TO FCB
7610 BD 71 27		JSR	GETFIL	GET FILE NAME
7613 25 54		BCS	ERROR6	ERROR?
7615 86 01		LDA R	#1	SET EXTENSION
7617 A7 00		STA R	0,1	
7619 BD 71 2D		JSR	SETEXT	
761C CE 77 40		LDX	#FCB	
761F BD 78 06		JSR	FMS	OPEN FILE
7622 26 26		BNE	WORD6	ERROR?
7624 BD 78 06	WORD2	JSR	FMC	GET CHARACTER
7627 26 21		BNE	WORD6	ERROR?

7629 81 0D		CMP A #\$D	IS IT CR?
762B 27 6A		BEQ WORD9	
762D 81 20		CMP A #\$20	IS IT SPACE?
762F 26 F3		BNE WORD2	
7631 BD 78 06	WORD4	JSR FMS	READ NEXT CHARACTER
7634 26 14		BNE WORD6	ERROR?
7636 81 0D		CMP A #\$D	IS IT CR?
7638 27 5D		BEQ WORD9	
763A 81 20		CMP A #\$20	IS IT SPACE?
763C 27 F3		BEQ WORD4	
763E FE 76 03		LDX COUNT	GET COUNT
7641 08		INX	INC BY ONE
7642 FF 76 03		STX COUNT	SVE COUNT
7645 CE 77 40		LDX #FCB	POINT TO FCB
7648 20 DA		BRA WORD2	REPEAT
764A A6 01	WORD6	LDA A 1,X	GET ERROR NUMBER
764C 81 08		CMP A #8	IS IT EOF?
764E 27 25		BEQ WORD8	
7650 A6 01	ERROR	LDA A 1,X	GET ERROR TYPE
7652 81 03		CMP A #3	FILE EXISTS?
7654 27 09		BEQ ERROR2	
7656 81 04		CMP A #4	NO FILE?
7658 27 0A		BEQ ERROR4	
765A BD 71 3C		JSR RPTERR	REPORT ERROR
765D 20 10		BRA WORD7	
765F CE 76 EC	ERROR2	LDX #FEST	POINT TO STRING
7662 20 08		BRA ERROR7	
7664 CE 76 CD	ERROR4	LDX #NST	POINT TO STRING
7667 20 03		BRA ERROR7	
7669 CE 76 DA	ERROR6	LDX #ILST	POINT TO STRING
766C BD 71 18	ERROR7	JSR PSTRNG	PRINT STRING
766F BD 78 03	WORD7	JSR FMSCLS	CLOSE ALL FILES
7672 7E 71 03		JMP WARMS	RETURN TO FLEX
7675 BD 71 1E	WORD8	JSR PCRLF	PRINT CR & LF
7678 CE 76 A3		LDX #STR	POINT TO STRING
767B BD 71 18		JSR PSTRNG	PRINT IT
767E CE 76 03		LDX #COUNT	POINT TO COUNT
7681 5F		CLR B	CLEAR FLAG
7682 BD 71 33		JSR OUTDEC	PRINT WORD COUNT
7685 CE 76 B9		LDX #STR2	POINT TO STRING
7688 BD 71 18		JSR PSTRNG	PRINT IT
768B 5F		CLR B	CLEAR FLAG
768C CE 76 05		LDX #LCOUNT	POINT TO LINE COUNT
768F BD 71 33		JSR OUTDEC	PRINT IT
7692 BD 71 1E		JSR PCRLF	PRINT CR & LF
7695 20 D8		BRA WORD7	
7697 FE 76 05	WORD9	LDX LCOUNT	POINT TO LINE COUNT
769A 08		INX	INC BY ONE
769B FF 76 05		STX LCOUNT	SAVE IT
769E CE 77 40		LDX #FCB	POINT TO FCB
76A1 20 8E		BRA WORD4	REPEAT

76A3 54	STR	FCC	'TOTAL WORD COUNT IS '
76B7 04		FCB	4
76B8 54	STR2	FCC	'TOTAL LINE COUNT IS '
76C0 04		FCB	4
76CD 4E	NSST	FCC	'NO SUCH FILE'
76D9 04		FCB	4
76DA 49	ILST	FCC	'ILLEGAL FILE NAME'
76EB 04		FCB	4
76EC 46	FEST	FCC	'FILE EXISTS'
76F7 04		FCB	4
		END	WORDS

NO ERROR(S) DETECTED

SYMBOL TABLE:

COUNT	7603	ERROR	7650	ERROR2	765F	ERROR4	7664	ERROR6	7669
ERROR7	766C	FCB	7740	FEST	76EC	FMS	7806	FMSCLS	7803
GETCHR	710F	GETFIL	7127	ILST	76DA	LCOUNT	7605	NSST	76CD
OUTDEC	7133	PCRLF	711E	PSTRNG	7118	PUTCHR	7112	RPTERR	713C
SETEXT	712D	STR	76A3	STR2	76B8	VN	7602	WARMS	7103
WORD0	7607	WORD2	7624	WORD4	7631	WORD6	764A	WORD7	766F
WORD8	7675	WORD9	7697	WORDS	7600				

TYPOS

The TYPOS utility is used for grouping and counting all words used in a text file. It is a great aid in detecting misspelled words and typographical errors, as well as pointing out too often used words in a document.

DESCRIPTION

The general syntax of the TYPOS command is:

```
TYPOS,<file spec>[,<count>]
```

where <count> specifies the highest word use count to be listed in the final word list. The file spec defaults to a TXT extension and to the working drive. If the count is not specified, the default will be 3, so words appearing three times or less will be listed. All letters are mapped to lower case so words like 'Test' and 'test' would be considered identical. The final word list is printed with each word preceded by its occurrence count and the word in lower case letters. The more often used words are printed first. Following are a few examples.

```
+++TYPOS,CHAPTER1  
+++TYPOS,BOOK,50
```

The first example would print all words occurring three times or less found in the file named CHAPTER1.TXT on the working drive. The second example would print all words occurring 50 times or less in the file BOOK.TXT. It should be noted that on long files, TYPOS may require a moderate amount of time to compile the list of words.

*
 * FIND [TYPOS] UTILITY
 *
 *
 * COPYRIGHT (C) 1978 BY
 *
 * TECHNICAL SYSTEMS CONSULTANTS, INC.
 * P. O. BOX 2574
 * WEST LAFAYETTE, INDIANA 47906
 *
 *

0100		ORG	\$0100	\$A100
7806	FMS	EQU	\$7806	\$B406
7103	WARMIS	EQU	\$7103	\$AD03
7740	FCB	EQU	\$7740	\$A040
7112	PUTCHR	EQU	\$7112	\$AD10
711B	CLASS	EQU	\$711B	\$AD21
7118	PSTRNG	EQU	\$7118	\$AD1E
711E	PCRLF	EQU	\$711E	\$AD24
7133	OUTDEC	EQU	\$7133	\$AD39
713C	RPTERR	EQU	\$713C	\$AD3F
7127	GETFIL	EQU	\$7127	\$AD20
712D	SETTEXT	EQU	\$712D	\$AD33
713F	GETHEX	EQU	\$713F	\$AD42
7121	NXTCH	EQU	\$7121	\$AD27
7803	FMSCLS	EQU	\$7803	\$B403
0100 20 2D	TYP	BRA	TYPO	
0102 01	VN	FCB	1	VERSION NUMBER
0103 00 00	DSNUM	FDB	0	
0105 00 00	TEMP	FDB	0	
0107 00 00	POINT	FDB	0	
0109 00 00	END	FDB	0	
010B 00 00	NUM	FDB	0	
010D 00 00	MAX	FDB	0	
010F 00 00	BUFFER	FDB	0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0	
012F CE 03 67	TYPO	LDX	#LIST	POINT TO LIST SPACE
0132 FF 01 09		STX	END	SET END
0135 CE 77 40		LDX	#FCB	POINT TO FCB
0138 BD 71 27		JSR	GETFIL	GET FILE SPEC
013B 25 68		BCS	TYP4	ERROR?
013D 86 01		LDA R	#1	SET FOR TEXT FILE
013F A7 00		STA R	0, X	SET FOR READ
0141 BD 71 2D		JSR	SETTEXT	SET EXTENSION
0144 CE 77 40		LDX	#FCB	
0147 BD 78 06		JSR	FMS	CALL FMS - OPEN
014A 26 68		BNE	TYP6	ERRORS?
014C BD 02 E5		JSR	INDEC	GET NUMBER

014F 25 57		BCS	TYP4	ERROR?
0151 FF 01 0B		STX	NUM	SAVE VALUE
0154 CE 01 0F	TYP1	LDX	#BUFFER	POINT TO WORD BUF
0157 FF 01 07		STX	POINT	SAVE POINTER
0158 CE 77 40	TYP2	LDX	#FCB	POINT TO FCB
015D BD 78 06		JSR	FMS	GET CHARACTER
0160 26 76		BNE	TYP?	ERROR?
0162 BD 71 1B		JSR	CLASS	CLASSIFY IT
0165 25 F3		BCS	TYP2	TERM?
0167 81 39		CMP A	#'9	IS IT NUMBER?
0169 22 0F		BHI	TYP35	
016B CE 77 40	TYP3	LDX	#FCB	POINT TO FCB
016E BD 78 06		JSR	FMS	GET CHARACTER
0171 26 65		BNE	TYP?	ERROR?
0173 BD 71 1B		JSR	CLASS	CLASSIFY IT
0176 25 E2		BCS	TYP2	
0178 20 F1		BRA	TYP3	
017A FE 01 07	TYP35	LDX	POINT	RESTORE POINTER
017D 81 41		CMP A	#\$41	CHECK FOR LETTER
017F 25 06		BLO	TYP37	
0181 81 5A		CMP A	#\$5A	
0183 22 02		BHI	TYP37	
0185 8B 20		ADD A	#\$20	MAKE IT LOWER CASE
0187 A7 00	TYP37	STA A	0, X	SAVE CHARACTER
0189 08		INX		BUMP TO NEXT
018A FF 01 07		STX	POINT	SAVE NEW POS
018D CE 77 40		LDX	#FCB	POINT TO FCB
0190 BD 78 06		JSR	FMS	GET CHARACTER
0193 26 43		BNE	TYP?	ERROR?
0195 BD 71 1B		JSR	CLASS	CLASSIFY IT
0198 24 E0		BCC	TYP35	
019A 81 2D		CMP A	#"~	IS IT HYPHEN?
019C 27 DC		BEQ	TYP35	
019E FE 01 07		LDX	POINT	GET POINTER
01A1 6F 00		CLR	0, X	SET END OF WORD
01A3 BD 02 54		JSR	INSERT	GO PUT IN LIST
01A6 20 AC		BRA	TYP1	REPEAT
01A8 CE 03 2F	TYP4	LDX	#SYST	POINT TO STRING
01AB BD 71 18		JSR	PSTRNG	PRINT IT
01AE BD 71 1E	TYP5	JSR	PCRLF	OUTPUT CR & LF
01B1 7E 71 03		JMP	WARM	RETURN TO FLEX
01B4 A6 01	TYP6	LDA A	1, X	GET ERROR TYPE
01B6 81 03		CMP A	#3	FILE EXISTS?
01B8 27 09		BEQ	ERROR2	
01BA 81 04		CMP A	#4	NO FILE?
01BC 27 0A		BEQ	ERROR4	
01BE BD 71 3C		JSR	RPTERR	REPORT ERROR
01C1 20 10		BRA	ERROR8	
01C3 CE 03 5B	ERROR2	LDX	#TEST	POINT TO STRING
01C6 20 08		BRA	ERROR7	
01C8 CE 03 3C	ERROR4	LDX	#INST	POINT TO STRING
01CB 20 03		BRA	ERROR7	
01CD CE 03 49	ERROR6	LDX	#LST	POINT TO STRING
01D0 BD 71 18	ERROR7	JSR	PTRNG	PRINT STRING

01D3 BD 78 03	ERROR8	JSR	FMSCLS	CLOSE ALL
01D6 20 D6		BRA	TYP5	
01D8 A6 01	TYP7	LDA A	1,X	GET ERROR TYPE
01DA 81 08		CMP A	#8	IS IT EOF?
01DC 26 D6		BNE	TYP6	

* PRINT WORD LIST

01DE BD 71 1E	PRINT	JSR	PCRLF	PRINT CR & LF
01E1 7D 01 0B		TST	NUM	NUMBER SET?
01E4 26 0A		BNE	PRIN05	
01E6 7D 01 0C		TST	NUM+1	
01E9 26 05		BNE	PRIN05	
01EB 86 03		LDA A	#3	SET DEFAULT 3
01ED B7 01 0C		STA A	NUM+1	
01F0 B6 01 0D	PRIN05	LDA A	MAX	CHECK MAX VALUE
01F3 F6 01 0E		LDA B	MAX+1	
01F6 B1 01 0B		CMP A	NUM	COMPARE TO NUMBER
01F9 22 0D		BHI	PRINT0	
01FB 25 05		BLO	PRINT07	
01FD F1 01 0C		CMP B	NUM+1	CHECK LOW HALF
0200 24 06		BHS	PRINT0	
0202 B7 01 0B	PRIN07	STA A	NUM	SET NEW VALUE
0205 F7 01 0C		STA B	NUM+1	
0208 CE 03 67	PRINT0	LDX	#LIST	POINT TO LIST
020B BC 01 09	PRINT1	CPX	END	END OF LIST?
020E 27 38		BEQ	PRINT6	
0210 FF 01 07		STX	POINT	SAVE POINTER
0213 60 00	PRIN15	TST	0,X	END OF WORD?
0215 27 03		BEQ	PRIN17	
0217 08		INX		BUMP TO NEXT
0218 20 F9		BRA	PRIN15	
021A 08	PRIN17	INX		
021B B6 01 0B		LDA A	NUM	GET NUMBER
021E F6 01 0C		LDA B	NUM+1	
0221 A1 00		CMP A	0,X	COMPARE COUNTS
0223 26 1F		BNE	PRINT5	
0225 E1 01		CMP B	1,X	
0227 26 18		BNE	PRINT5	
0229 BD 71 1E	PRIN18	JSR	PCRLF	OUTPUT CR & LF
022C C6 01		LDA B	#1	SET FLAG
022E BD 71 33		JSR	OUTDEC	OUTPUT COUNT
0231 FE 01 07		LDX	POINT	RESTORE POINTER
0234 86 20		LDA A	#\$20	OUTPUT SPACE
0236 BD 71 12		JSR	PUTCHR	
0239 A6 00	PRINT3	LDA A	0,X	GET CHARACTER
023B 27 06		BEQ	PRINT4	NULL?
023D BD 71 12		JSR	PUTCHR	PRINT CHARACTER
0240 08		INX		BUMP TO NEXT
0241 20 F6		BRA	PRINT3	
0243 08	PRINT4	INX		
0244 08	PRINT5	INX		MOVE TO NEXT WORD
0245 08		INX		
0246 20 C3		BRA	PI INT1	REPEAT

0248 FE 01 0B	PRINT6	LDX	NUM	GET NUMBER
024B 09		DEX		DEC IT
024C FF 01 0B		STX	NUM	SAVE NEW
024F 26 B7		BNE	PRINT0	FINISHED?
0251 7E 01 AE		JMP	TYP5	

* INSERT WORD IN LIST

0254 CE 03 67	INSERT	LDX	#LIST	POINT TO LIST
0257 FF 01 05	INS2	STX	TEMP	SAVE IT
025A CE 01 0F		LDX	#BUFFER	POINT TO WORD
025D FF 01 07		STX	POINT	
0260 FE 01 05		LDX	TEMP	
0263 BC 01 09		CPX	END	END OF LIST?
0266 27 30		BEQ	ADD	
0268 FE 01 07	INS3	LDX	POINT	SET POINTER
026B A6 00		LDA A	0,X	GET CHARACTER
026D 26 09		BNE	INS35	NULL?
026F FE 01 05		LDX	TEMP	RESET POINTER
0272 6D 00		TST	0,X	TEST FOR END
0274 27 51		BEQ	MATCH	IF SO, MATCH!
0276 20 14		BRA	INS4	REPEAT
0278 08	INS35	INX		BUMP TO NEXT
0279 FF 01 07		STX	POINT	SAVE POSITION
027C FE 01 05		LDX	TEMP	
027F 6D 00		TST	0,X	TEST FOR END
0281 27 10		BEQ	INS5	
0283 08		INX		
0284 FF 01 05		STX	TEMP	SAVE POINTER
0287 09		DEX		
0288 A1 00		CMP A	0,X	CHECK CHARACTER
028A 27 DC		BEQ	INS3	EQUAL?
028C 6D 00	INS4	TST	0,X	CHECK FOR END
028E 27 03		BEQ	INS5	
0290 08		INX		
0291 20 F9		BRA	INS4	REPEAT
0293 08	INS5	INX		
0294 08		INX		MOVE TO NEXT WORD
0295 08		INX		
0296 20 BF		BRA	INS2	REPEAT

* ADD WORD TO LIST

0298 FE 01 09	ADD	LDX	END	POINT TO END
029B FF 01 05		STX	TEMP	
029E CE 01 0F		LDX	#BUFFER	POINT TO WORD
02A1 FF 01 07		STX	POINT	
02A4 A6 00	ADD2	LDA A	0,X	GET CHARACTER
02A6 08		INX		BUMP TO NEXT
02A7 FF 01 07		STX	POINT	
02AA FE 01 05		LDX	TEMP	POINT TO LIST
02AD A7 00		STA A	0,X	SAVE CHARACTER
02AF 27 09		BEQ	ADD4	END?
02B1 08		INX		BUMP TO NEXT

02B2 FF 01 05		STX	TEMP	SAVE POSITION
02B5 FE 01 07		LDX	POINT	
02B8 20 EA		BRA	ADD2	REPEAT
02BA 08	ADD4	INX		
02BB 6F 00		CLR	0,X	SET COUNT
02BD 86 01		LDA A	#1	
02BF A7 01		STA A	1,X	
02C1 08		INX		SET END OF LIST
02C2 08		INX		
02C3 FF 01 09		STX	END	SAVE IT
02C6 39		RTS		

* WORD FOUND

02C7 08	MATCH	INX		
02C8 6C 01		INC	1,X	INC COUNT
02CA 26 02		BNE	MATCH2	
02CC 6C 00		INC	0,X	
02CE R6 00		LDA A	0,X	GET COUNT
02D0 E6 01		LDA B	1,X	
02D2 B1 01 0D	CMP A	MAX	CMPR TO MAXIMUM	
02D5 25 0D	BLO	MATCH6		
02D7 22 05	BHI	MATCH4		
02D9 F1 01 0E	CMP B	MAX+1	CHECK LOW HALF	
02DC 23 06	BLS	MATCH6		
02DE B7 01 0D	STA A	MAX	SET NEW MAX	
02E1 F7 01 0E	STA B	MAX+1		
02E4 39	MATCH6	RTS	RETURN	

* INPUT DECIMAL NUMBER

02E5 7F 01 03	INDEC	CLR	DSNUM	CLEAR VALUE
02E8 7F 01 04		CLR	DSNUM+1	
02EB 5F		CLR B		CLEAR COUNTER
02EC BD 71 21		JSR	NXTCH	GET CHARACTER
02EF 25 33		BCS	INDEC4	FINISHED?
02F1 81 39		CMP A	#'9	IS IT NUMBER?
02F3 22 34		BHI	INDEC6	ERROR?
02F5 84 0F		AND A	#\$F	MASK VALUE
02F7 37		PSH B		SAVE COUNT
02F8 36		PSH A		SAVE NUMBER
02F9 B6 01 03	LDA A	DSNUM	GET NUMBER	
02FC F6 01 04	LDA B	DSNUM+1		
02FF 58	ASL B		MULT BY 6	
0300 49	ROL A			
0301 58	ASL B			
0302 49	ROL A			
0303 58	ASL B			
0304 49	ROL A			
0305 78 01 04	ASL	DSNUM+1	TIMES 2	
0308 79 01 03	ROL	DSNUM		
0308 FB 01 04	ADD B	DSNUM+1	ADD IN NEW	
030E B9 01 03	ADC A	DSNUM		
0311 F7 01 04	STA B	DSNUM+1	SAVE NEW	

0314 33		PUL B		
0315 FB 01 04		ADD B	DSNUM+1	
0318 89 00		ADC A	#0	
031A F7 01 04		STR B	DSNUM+1	SAVE RESULT
031D B7 01 03		STR A	DSNUM	
0320 33		PUL B		RESET COUNTER
0321 5C		INC B		BUMP THE COUNT
0322 20 C8		BRA	INDEC2	REPEAT
0324 FE 01 03	INDEC4	LDX	DSNUM	GET VALUE
0327 0C		CLC		CLEAR ERRORS
0328 39		RTS		
0329 BD 71 21	INDEC6	JSR	NXTCH	GET NEXT CHAR
032C 24 FB		BCC	INDEC6	TERM?
032E 39		RTS		EXIT WITH ERROR

* STRINGS

032F 53	SYST	FCC	'SYNTAX ERROR'
033B 04		FCB	4
033C 4E	NSST	FCC	'NO SUCH FILE'
0348 04		FCB	4
0349 49	ILST	FCC	'ILLEGAL FILE NAME'
035A 04		FCB	4
035B 46	FEST	FCC	'FILE EXISTS'
0366 04		FCB	4
0367	LIST	EQU	*
		END	TYP

NO ERROR(S) DETECTED

SYMBOL TABLE:

ADD 0298	ADD2 02A4	ADD4 02BA	BUFFER 010F	CLASS 711B
DSNUM 0103	END 0109	ERROR2 01C3	ERROR4 01C8	ERROR6 01CD
ERROR7 01D0	ERROR8 01D3	FCB 7740	FEST 035B	FMS 7806
FMSCLS 7803	GETFIL 7127	GETHEX 713F	ILST 0349	INDEC 02E5
INDEC2 02EC	INDEC4 0324	INDEC6 0329	INS2 0257	INS3 0268
INS35 0278	INS4 028C	INS5 0293	INSERT 0254	LIST 0367
MATCH 02C7	MATCH2 02CE	MATCH4 02DE	MATCH6 02E4	MAX 01BD
NSST 033C	NUM 010B	NXTCH 7121	OUTDEC 7133	PCRLF 711E
POINT 0107	PRIN05 01F0	PRIN07 0202	PRIN15 0213	PRIN17 021A
PRIN18 0229	PRINT 01DE	PRINT0 0208	PRINT1 020B	PRINT3 0239
PRINT4 0243	PRINT5 0244	PRINT6 0248	PSTRNG 7118	PUTCHR 7112
RPTERR 713C	SETEXT 712D	SYST 032F	TEMP 0105	TYP 0108
TYPO 012F	_TYP1 0154	TYP2 015A	TYP3 016B	TYP35 017H
TYP37 0187	TYP4 01A8	TYP5 01AE	TYP6 01B4	TYP7 01D8
VN 0102	WARMS 7103			

SPLIT

The SPLIT command is used to split a text file into two new files at a specified line number. It is convenient to use when a file becomes too large to easily manage or to break off an often used section of text into another file.

DESCRIPTION

The general syntax of the SPLIT command is:

```
SPLIT,<input file spec>,<out file spec1>,<out file spec2>,<N>
```

The input file is the file to be split, output file spec 1 is the name to be assigned to the first set of lines read from the input file, output spec 2 is the name to be assigned to the rest of the file being split, and N is the line number where the file should be split. The second output file will begin with line N of the input file. All files default to TXT extensions and to the working drive. An example follows:

```
+++SPLIT,TEST,TEST1,TEST2,125
```

This command line would cause lines 1 to 124 of the file named TEST.TXT on the working drive to be written into a file named TEST1.TXT and lines 125 to the end of the file to be written into a file named TEST2.TXT. The original file (TEST) remains unchanged.

*
 * [SPLIT] FILE UTILITY
 *
 *
 * COPYRIGHT (C) 1978 BY
 *
 * TECHNICAL SYSTEMS CONSULTANTS, INC.
 * P. O. BOX 2574
 * WEST LAFAYETTE, INDIANA 47906
 *
 *

* EQUATES

7127	GETFIL	EQU	\$7127	\$1D2D
712D	SETEXT	EQU	\$712D	\$1D33
7740	FCB	EQU	\$7740	\$1840
7806	FMS	EQU	\$7806	\$B406
7803	FMSCLS	EQU	\$7803	\$B403
713C	RPTERR	EQU	\$713C	\$1D3F
7103	WARM8	EQU	\$7103	\$A003
7118	PSTRNG	EQU	\$7118	\$A01E
7121	NXTCH	EQU	\$7121	\$1D27
0100		ORG	\$0100	<u>\$0100</u>

* PROGRAM STARTS HERE

0100 20 07 SPLIT BRA SPLIT1

* TEMPS

0102 01	VN	FCB	1	VERSION NUMBER
0103 00 00	DSNUM	FDB	0	
0105 00 00	LINE	FDB	0	
0107 00 00	SLINE	FDB	0	

0109 CE 77 40	SPLIT1	LDX	#FCB	POINT TO FCB
010C BD 71 27		JSR	GETFIL	GET FILE SPEC
010F 25 6F		BCS	ERROR6	ERROR?
0111 86 01		LDA A	#1	SET FOR TXT
0113 A7 00		STA A	0,X	READY FOR OPEN
0115 BD 71 2D		JSR	SETEXT	SET EXTENSION
0118 CE 77 40		LDX	#FCB	
011B BD 78 06		JSR	FMS	CALL FMS - OPEN
011E 26 47		BNE	SPLIT2	ERROR?
0120 CE 02 65		LDX	#FCB1	POINT TO FCB
0123 BD 71 27		JSR	GETFIL	GET FILE SPEC
0126 25 58		BCS	ERROR6	ERROR?
0128 86 01		LDA A	#1	TXT CODE
012A BD 71 2D		JSR	SEIEXT	SET EXTENSION
012D 86 02		LDA A	#2	OPEN FOR WRITE
012F CE 02 65		LDX	#FCB1	

0132 A7 00	STA A	0,X	
0134 BD 78 06	JSR	FMS	CALL FMS - OPEN
0137 26 2E	BNE	SPLIT2	ERRORS?
0139 CE 03 25	LDX	#FCB2	POINT TO FCB
013C BD 71 27	JSR	GETFIL	GET FILE SPEC
013F 25 3F	BCS	ERROR6	
0141 86 01	LDA A	#1	SET FOR TXT
0143 BD 71 2D	JSR	SETEXT	SET EXT
0146 CE 03 25	LDX	#FCB2	
0149 86 02	LDA A	#2	OPEN FOR WRITE
014B A7 00	STA A	0,X	
014D BD 78 06	JSR	FMS	CALL FMS - OPEN
0150 26 15	BNE	SPLIT2	ERRORS?
0152 BD 01 D2	JSR	INDEC	GET LINE NUMBER
0155 25 35	BCS	SPLIT4	ERROR?
0157 5D	TST B		NUMBER THERE?
0158 27 32	BEQ	SPLIT4	
015A FF 01 07	STX	SLINE	SAVE NUMBER
015D 27 2D	BEQ	SPLIT4	
015F CE 00 01	LDX	#1	INIT LINE COUNT
0162 FF 01 05	STX	LINE	
0165 20 2D	BRA	SPLIT6	
0167 A6 01	SPLIT2	LDA A	GET ERROR TYPE
0169 81 03		CMP A	FILE EXISTS?
016B 27 09	BEQ	ERROR2	
016D 81 04		CMP A	NO FILE?
016F 27 0A	BEQ	ERROR4	
0171 BD 71 3C	JSR	RPTERR	REPORT ERROR
0174 20 10	BRA	SPLIT3	
0176 CE 02 59	ERROR2	LDX	POINT TO STRING
0179 20 08	BRA	ERROR7	
017B CE 02 3A	ERROR4	LDX	POINT TO STRING
017E 20 03	BRA	ERROR7	
0180 CE 02 47	ERROR6	LDX	POINT TO STRING
0183 BD 71 18	ERROR7	JSR	PRINT STRING
0186 BD 78 03	SPLIT3	JSR	CLOSE ALL FILES
0189 7E 71 03		JMP	WARMS
018C CE 02 1C	SPLIT4	LDX	RETURN TO FLEX
018F BD 71 18		JSR	POINT TO STRING
0192 20 F2	BRA	SPLIT3	PRINT IT
0194 FE 01 05	SPLIT6	LDX	GET COUNT
0197 BC 01 07		CPX	SPLIT YET?
019A 27 24	BEQ	SPLIT8	
019C 08		INX	BUMP COUNT
019D FF 01 05		STX	LINE
01A0 CE 77 40	SPLIT65	LDX	POINT TO FCB
01A3 BD 78 06		JSR	READ CHARACTER
01A6 26 10	BNE	SPLIT7	ERROR?
01A8 CE 02 65		LDX	POINT TO FCB
01AB 36	PSH A	#FCB1	SAVE CHAR
01AC BD 78 06	JSR	FMS	WRITE CHARACTER
01AF 32		PUL A	RESTORE
01B0 26 B5	BNE	SPLIT2	ERRORS?
01B2 81 0D		CMP A	WAS IT CR?

01B4 27 DE		BEQ	SPLIT6	
01B6 20 E8		BRA	SPLIT5	
01B8 A6 01	SPLIT7	LDA A	1,X	GET ERROR TYPE
01BA 81 08		CMP A	#8	IS IT EOF?
01BC 26 A9		BNE	SPLIT2	
01BE 20 C6		BRA	SPLIT3	
01C0 CE 77 40	SPLIT8	LDX	#FCB	POINT TO FCB
01C3 BD 78 06		JSR	FMS	READ CHARACTER
01C6 26 F0		BNE	SPLIT7	ERROR?
01C8 CE 03 25		LDX	#FCB2	POINT TO FCB
01CB BD 78 06		JSR	FMS	WRITE CHARACTER
01CE 26 97		BNE	SPLIT2	ERROR?
01D0 20 EE		BRA	SPLIT8	

* INPUT DECIMAL NUMBER

01D2 7F 01 03	INDEC	CLR	DSNUM	CLEAR VALUE
01D5 7F 01 04		CLR	DSNUM+1	
01D8 5F		CLR B		CLEAR COUNTER
01D9 BD 71 21	INDEC2	JSR	NXTCH	GET CHARACTER
01DC 25 33		BCS	INDEC4	FINISHED?
01DE 81 39		CMP A	#19	IS IT NUMBER?
01E0 22 34		BHI	INDEC6	ERROR?
01E2 84 0F		AND A	#\$F	MASK VALUE
01E4 37		PSH B		SAVE COUNT
01E5 36		PSH A		SAVE NUMBER
01E6 B6 01 03		LDA A	DSNUM	GET NUMBER
01E9 F6 01 04		LDA B	DSNUM+1	
01EC 58		ASL B		MULT BY 6
01ED 49		ROL A		
01EE 58		ASL B		
01EF 49		ROL A		
01F0 58		ASL B		
01F1 49		ROL A		
01F2 78 01 04		ASL	DSNUM+1	TIMES 2
01F5 79 01 03		ROL	DSNUM	
01F8 FB 01 04		ADD B	DSNUM+1	ADD IN NEW
01F8 B9 01 03		ADC A	DSNUM	
01FE F7 01 04		STA B	DSNUM+1	SAVE NEW
0201 33		PUL B		
0202 FB 01 04		ADD B	DSNUM+1	
0205 89 00		RDC A	#0	
0207 F7 01 04		STA B	DSNUM+1	SAVE RESULT
0208 B7 01 03		STA A	DSNUM	
020D 33		PUL B		RESET COUNTER
020E 5C		INC B		BUMP THE COUNT
020F 20 C8		BRA	INDEC2	REPEAT
0211 FE 01 03	INDEC4	LDX	DSNUM	GET VALUE
0214 0C		CLC		CLEAR ERRORS
0215 39		RTS		
0216 BD 71 21	INDEC6	JSR	NXTCH	GET NEXT CHAR
0219 24 FB		BCC	INDEC6	TERM?
021B 39		RTS		EXIT WITH ERROR

* STRINGS

021C 49	INST	FCC	'ILLEGAL LINE NUMBER SPECIFIED'
0239 04		FCB	4
023A 4E	NSST	FCC	'NO SUCH FILE'
0246 04		FCB	4
0247 49	ILST	FCC	'ILLEGAL FILE NAME'
0258 04		FCB	4
0259 46	FEST	FCC	'FILE EXISTS'
0264 04		FCB	4

* FCBS

0265	FCB1	RMB	192	320
0325	FCB2	RMB	192	320
		END	SPLIT	

NO ERROR(S) DETECTED

SYMBOL TABLE:

DSNUM 0103	ERROR2 0176	ERROR4 017B	ERROR6 0180	ERROR7 0183
FCB 7740	FCB1 0265	FCB2 0325	FEST 0259	FMS 7806
FMSCLS 7803	GETFIL 7127	ILST 0247	INDEC 01D2	INDEC2 01D9
INDEC4 0211	INDEC6 0216	INST 021C	LINE 0105	NSST 023A
NXTCH 7121	PSTRNG 7118	RPTERR 713C	SETEXT 712D	SLINE 0107
SPLIT5 01A0	SPLIT 0100	SPLIT1 0109	SPLIT2 0167	SPLIT3 0186
SPLIT4 018C	SPLIT6 0194	SPLIT7 01B8	SPLIT8 01C0	VN 0102
WARMSS 7103				

LOW-UP

The LOW-UP command is used to convert a file into all upper case letters. It is useful for those systems unable to work with lower case letters.

DESCRIPTION

The general syntax of the LOW-UP command is:

```
LOW-UP,<input file spec>,<output file spec>
```

The input file spec specifies the name of the file needing the conversion, and the output file spec specifies the file name of the new converted file. Both default to a TXT extension and to the working drive. The new file will end up containing only upper case letters and the original file will be left unchanged. An example follows:

```
+++LOW-UP,LISTER,LISTERU
```

This would cause a file named LISTERU.TXT to be created which is identical to the file named LISTER.TXT except all letters will be upper case.

```

*
* LOWER TO UPPER CASE CONVERTER
*
*
* COPYRIGHT (C) 1978 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*

```

* MAIN DISK EQUATES

7740	FCB	EQU	\$7740	/A040
7806	FMS	EQU	\$7806	/B406
7803	FMSCLS	EQU	\$7803	/B403
7103	WARM5	EQU	\$7103	/A003
710F	GETCHR	EQU	\$710F	/AD15
7112	PUTCHR	EQU	\$7112	/AD18
7118	PSTRNG	EQU	\$7118	/AD1E
711E	PCRLF	EQU	\$711E	/AD24
7127	GETFIL	EQU	\$7127	/AD2D
712D	SETEXT	EQU	\$712D	/AD33
713C	RPTERR	EQU	\$713C	/AD3F
0100		ORG	\$0100	/A100

* MAIN STARTS HERE

0100 20 01	LOW	BRA	LOW1	
0102 01	VN	FCB	1	VERSION NUMBER
0103 CE 77 40	LOW1	LDX	#FCB	POINT TO FCB
0106 BD 71 27		JSR	GETFIL	GET THE FILE NAME
0109 25 28		BCS	ERROR6	
010B 86 01		LDA A	#1	SET FOR READ
010D A7 00		STA A	0,X	SAVE IN FCB
010F BD 71 2D		JSR	SETEXT	SET DEFAULT EXT
0112 CE 77 40		LDX	#FCB	
0115 BD 78 06		JSR	FMS	CALL FMS
0118 27 25		BEQ	LOW2	ERRORS?
011A A6 01	DSKERR	LDA A	1,X	GET ERROR TYPE
011C 81 03		CMP A	#3	FILE EXISTS?
011E 27 09		BEQ	ERROR2	
0120 81 04		CMP A	#4	NO FILE?
0122 27 0A		BEQ	ERROR4	
0124 BD 71 3C		JSR	RPTERR	REPORT ERROR
0127 20 10		BRA	DSKER2	
0129 CE 01 A8	ERROR2	LDX	#FEST	POINT TO STRING
012C 20 08		BRA	ENPOP7	

LOWER TO UPPER CASE CONVERTER

TSC MNEMONIC ASSEMBLER

PAGE 2

012E CE 01 89	ERROR4	LDX	#NSST	POINT TO STRING
0131 20 03		BRA	ERROR7	
0133 CE 01 96	ERROR6	LDX	#ILST	POINT TO STRING
0136 BD 71 18	ERROR7	JSR	PSTRNG	PRINT STRING
0139 BD 78 03	DSKERZ	JSR	FMSCLS	CLOSE ALL FILES
013C 7E 71 03		JMP	WARMs	RETURN TO FLEX
013F CE 01 B4	LOW2	LDX	#FCB2	POINT TO FCB
0142 BD 71 27		JSR	GETFIL	GET FILE NAME
0145 25 EC		BCS	ERROR6	ERROR?
0147 86 01		LDA A	#1	SET EXTENSION
0149 BD 71 2D		JSR	SETEXT	
014C CE 01 B4		LDX	#FCB2	
014F 86 02		LDA A	#2	OPEN FOR WRITE
0151 A7 00		STA A	0; X	
0153 BD 78 06		JSR	FMS	CALL FMS
0156 26 C2		BNE	DSKERR	ERROR?
0158 CE 77 40	LOW5	LDX	#FCB	POINT TO READ
015B BD 78 06		JSR	FMS	GET CHARACTER
015E 26 14		BNE	LOW7	
0160 81 61		CMP A	#'a	I IT LOWER CASE?
0162 25 06		BLO	LOW6	
0164 81 7A		CMP A	#'z	CHECK SOME MORE
0166 22 02		BHI	LOW6	
0168 80 20		SUB A	#\$20	MAKE UPPER!
016A CE 01 B4	LOW6	LDX	#FCB2	POINT TO WRITE
016D BD 78 06		JSR	FMS	WRITE CHARACTER
0170 26 A8		BNE	DSKERR	ERROR?
0172 20 E4		BRA	LOW5	REPEAT
0174 A6 01	LOW7	LDA A	1; X	CHECK ERROR
0176 81 08		CMP A	#8	IS IT EOF?
0178 26 A0		BNE	DSKERR	
017A CE 01 B4		LDX	#FCB2	POINT TO WRITE
017D 86 04		LDA A	#4	CLOSE FILE
017F A7 00		STA A	0; X	
0181 BD 78 06		JSR	FMS	CALL FMS
0184 26 94		BNE	DSKERR	ERROR?
0186 7E 71 03		JMP	WARMs	RETURN TO FLEX

*: STRINGS

0189 4E	NSST	FCC	'NO SUCH FILE'
0195 04		FCB	'4'
0196 49	ILST	FCC	'ILLEGAL FILE NAME'
01A7 04		FCB	'4'
01A8 46	FEST	FCC	'FILE EXISTS'
01B3 04		FCB	'4'
01B4	FCB2	RMB	192
		END	LOW

320

NO ERROR(S) DETECTED

UP-LOW

The UP-LOW command is used to convert a file into all lower case letters. It is useful for those systems unable to work with upper case letters or to make a file easier to read.

DESCRIPTION

The general syntax of the UP-LOW command is:

```
UP-LOW,<input file spec>,<output file spec>
```

The input file spec specifies the name of the file needing the conversion, and the output file spec specifies the file name of the new converted file. Both default to a TXT extension and to the working drive. The new file will end up containing only lower case letters and the original file will be left unchanged. An example follows:

```
+++UP-LOW,TEXTA,TEXTAL
```

This would cause a file named TEXTAL.TXT to be created which is identical to the file named TEXTA.TXT except all letters will be lower case.

```

*
* UPPER TO LOWER CASE CONVERTER.
*
*
* COPYRIGHT (C) 1978 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*

```

* MAIN DISK EQUATES

7740	FCB	EQU	\$7740	4A040
7806	FMS	EQU	\$7806	4B406
7803	FMSCLS	EQU	\$7803	4B403
7103	WARMS	EQU	\$7103	4A003
710F	GETCHR	EQU	\$710F	4A015
7112	PUTCHR	EQU	\$7112	4A010
7118	PSTRNG	EQU	\$7118	4A01E
711E	PCRLF	EQU	\$711E	4A024
7127	GETFIL	EQU	\$7127	4A02D
712D	SETEXT	EQU	\$712D	4A033
713C	RPTERR	EQU	\$713C	4A03F
0100		ORG	\$0100	4A100

* MAIN STARTS HERE

0100 20 01	UP	BRA	UP1	
0102 01	VN	FCB	1	VERSION NUMBER
0103 CE 77 40	UP1	LDX	#FCB	POINT TO FCB
0106 BD 71 27		JSR	GETFIL	GET THE FILE NAME
0109 25 28		BCS	ERROR6	
010B 86 01		LDA A	#1	SET FOR READ
010D A7 00		STA A	0,X	SAVE IN FCB
010F BD 71 2D		JSR	SETEXT	SET DEFAULT EXT
0112 CE 77 40		LDX	#FCB	
0115 BD 78 06		JSR	FMS	CALL FMS
0118 27 25		BEQ	UP2	
011A A6 01	DSKERR	LDA A	1,X	GET ERROR TYPE
011C 81 03		CMP A	#3	FILE EXISTS?
011E 27 09		BEQ	ERROR2	
0120 81 04		CMP A	#4	NO FILE?
0122 27 0A		BEQ	ERROR4	
0124 BD 71 3C		JSR	RPTERR	REPORT ERROR
0127 20 10		BRA	DSKERR	
0129 CE 01 A8	ERROR2	LDX	#FLST	POINT TO STRING
012C 20 08		BRA	ERROR7	

UPPER TO LOWER CASE CONVERTER

TSC MNEMONIC ASSEMBLER

PAGE 2

012E CE 01 89	ERROR4	LDX	#NSST	POINT TO STRING
0131 20 03		BRA	ERROR7	
0133 CE 01 96	ERROR6	LDX	#ILST	POINT TO STRING
0136 BD 71 18	ERROR7	JSR	PSTRNG	PRINT STRING
0139 BD 78 03	DSKER2	JSR	FMSCLS	CLOSE ALL FILES
013C 7E 71 03		JMP	WARMS	RETURN TO FLEX
013F CE 01 B4	UP2	LDX	#FCB2	POINT TO FCB
0142 BD 71 27		JSR	GETFIL	GET FILE NAME
0145 25 EC		BCS	ERROR6	ERROR?
0147 86 01		LDA A	#1	SET EXTENSION
0149 BD 71 2D		JSR	SETEXT	
014C CE 01 B4		LDX	#FCB2	
014F 86 02		LDA A	#2	OPEN FOR WRITE
0151 A7 00		STA A	0,X	
0153 BD 78 06		JSR	FMS	CALL FMS
0156 26 C2		BNE	DSKERR	ERROR?
0158 CE 77 40	UP5	LDX	#FCB	POINT TO READ
015B BD 78 06		JSR	FMS	GET CHARACTER
015E 26 14		BNE	UP7	
0160 81 41		CMP A	#'A	I IT UPPER CASE?
0162 25 06		BLO	UP6	
0164 81 5A		CMP A	#'Z	CHECK SOME MORE
0166 22 02		BHI	UP6	
0168 8B 20		ADD A	#\$20	MAKE UPPER!
016A CE 01 B4	UP6	LDX	#FCB2	POINT TO WRITE
016D BD 78 06		JSR	FMS	WRITE CHARACTER
0170 26 A8		BNE	DSKERR	ERROR?
0172 20 E4		BRA	UP5	REPEAT
0174 A6 01	UP7	LDA A	1,X	CHECK ERROR
0176 81 08		CMP A	#0	IS IT EOF?
0178 26 A0		BNE	DSKERR	
017A CE 01 B4		LDX	#FCB2	POINT TO WRITE
017D 86 04		LDA A	#4	CLOSE FILE
017F A7 00		STA A	0,X	
0181 BD 78 06		JSR	FMS	CALL FMS
0184 26 94		BNE	DSKERR	ERROR?
0186 7E 71 03		JMP	WARMS	RETURN TO FLEX

* STRINGS

0189 4E	NSST	FCC	'NO SUCH FILE'
0195 04		FCB	4
0196 49	ILST	FCC	'ILLEGAL FILE NAME'
01A7 04		FCB	4
01A8 46	FEST	FCC	'FILE EXISTS'
01B3 04		FCB	4
01B4	FCB2	RMB	192 320
		END	UP

NO ERROR(S) DETECTED

DUMP

The DUMP utility is used for dumping the contents of a file, one sector at a time, in both hex and ASCII characters. It can be used as a disk debugging aid or to clarify the exact format for disk files.

DESCRIPTION

The general syntax of the DUMP command is:

```
DUMP,<file spec>
```

where <file spec> specifies the file to be dumped and defaults to a BIN extension. As each sector is displayed it will be preceded by two, 2 digit numbers, the first being the hex value of the track number, the second being the sector number of the sector being dumped. Each data line will contain 16 hex digits representing the data followed by the ASCII representations of the data. All non-printable characters are displayed as underscores (_). An example follows:

```
+++DUMP,FILE55
```

This would cause the contents of each one of the sectors contained in the file named FILE55.BIN to be dumped on the output device.

```

*
* DUMP FILE IN HEX AND ASCII
*
*
* COPYRIGHT (C) 1978 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*

```

0100		ORG	\$0100	\$A10-C
7740	FCB	EQU	\$7740	\$A0240
7139	OUTHEX	EQU	\$7139	\$A103C
710F	GETCHR	EQU	\$710F	\$A105-
7127	GETFIL	EQU	\$7127	\$A1020
7112	PUTCHR	EQU	\$7112	\$A1010
7118	PSTRNG	EQU	\$7118	\$A10E
711E	PCRLF	EQU	\$711E	\$A1024
7806	FMS	EQU	\$7806	\$B406
7103	WARMS	EQU	\$7103	\$A10C3
713C	RPTERR	EQU	\$713C	\$A103F
712D	SETEXT	EQU	\$712D	\$A1033
7803	FMSCLS	EQU	\$7803	\$B403
0100 20 06	DUMP	BRA	DUMP1	
0102 01	VN	FCB	1	
0103 00	TRACK	FCB	0	
0104 00	SECTOR	FCB	0	
0105 00	PASS	FCB	0	
0106 00 06	TEMP	FDB	0	
0108 CE 77 40	DUMP1	LDX	#FCB	POINT TO FCB
010B BD 71 27		JSR	GETFIL	GET FILE NAME
010E 24 06		BCC	DUMP12	ERROR?
0110 7E 01 C4		JMP	ERROR6	
0113 7E 01 AB	DUMP11	JMP	READ2	REPORT ERROR
0116 4F	DUMP12	CLR A		SET EXTENSION
0117 BD 71 2D		JSR	SETEXT	
0118 CE 77 40		LDX	#FCB	
011D 86 01		LDA A	#1	OPEN FOR READ
011F A7 00		STA A	0,X	
0121 BD 78 06		JSR	FMS	CALL FMS
0124 26 ED		BNE	DUMP11	ERRORS?
0126 A6 11		LDA A	17,1	GET TRACK
0128 B7 01 03		STA A	TRACK	
012B A6 12		LDA A	18,1	GET SECTOR
012D B7 01 04		STA A	SECTOR	
0130 86 09		LDA A	#9	SET FOR SS READ
0132 A7 00		STA A	0,X	

DUMP FILE UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 2

0134 BD 01 98	LOOP	JSR	READ	GO DO READ
0137 BD 71 1E		JSR	PCRLF	OUTPUT CR & LF
013A CE 01 03		LDX	#TRACK	POINT TO TRACK
013D BD 01 90		JSR	OUTH	OUTPUT TRACK
0140 CE 01 04		LDX	#SECTOR	POINT TO SECTOR
0143 BD 01 90		JSR	OUTH	OUTPUT IT
0146 BD 71 1E		JSR	PCRLF	PRINT CR & LF
0149 CE 77 20		LDX	#FCB+64	POINT TO DATA
014C 86 09		LDA A	#8	SET LINE COUNT
014E B7 01 05		STA A	PASS	
0151 C6 10	DUMP2	LDR B	#16	SET COLUMN COUNT
0153 FF 01 06		STX	TEMP	SAVE IT
0156 37	DUMP25	PSH B		
0157 BD 01 90		JSR	OUTH	OUTPUT DATA BYTE
015A 08		INX		BUMP TO NEXT
015B 33		PUL B		
015C 5A		DEC B		DEC THE COUNT
015D 26 F7		BNE	DUMP25	
015F FE 01 06		LDX	TEMP	RESTORE POINTER
0162 C6 10		LDA B	#16	RESET COUNT
0164 R6 00	DUMP27	LDA A	0,X	GET CHARACTER
0166 84 7F		AND A	#\$7F	MASK MSB
0168 81 1F		CMP A	#\$1F	IS IT CONTROL?
016A 22 02		BHI	DUMP28	
016C 86 5F		LDA A	#'_	IF SO, OUTPUT '_'
016E BD 71 12	DUMP28	JSR	PUTCHR	OUTPUT CHARACTER
0171 08		INX		BUMP TO NEXT
0172 5A		DEC B		DEC THE COUNT
0173 26 EF		BNE	DUMP27	
0175 BD 71 1E		JSR	PCRLF	OUTPUT CR & LF
0178 7A 01 05		DEC	PASS	DEC LINE COUNT
017B 26 D4		BNE	DUMP2	
017D B6 77 80		LDA A	FCB+64	GET TRACK
0180 27 0B		BEP	DUMP6	END OF FILE?
0182 B7 01 03		STA A	TRACK	SAVE IT
0185 B6 77 81		LDA A	FCB+65	GET SECTOR
0188 B7 01 04		STA A	SECTOR	SAVE IT
018B 20 A7		BRA	LOOP	REPEAT
018D 7E 71 03	DUMP6	JMP	WARMS	RETURN TO FLEX

* OUTPUT HEX DIGIT

0190 BD 71 39	OUTH	JSR	OUTHEX	OUTPUT IT
0193 86 20		LDA A	#\$20	SETUP SPACE
0195 7E 71 12		JMP	PUTCHR	OUTPUT IT

* READ NEXT SECTOR

0198 CE 77 40	READ	LDX	#FCB	POINT TO FCB
019B B6 01 03		LDA A	TRACK	GET TRACK
019E A7 1E		STA A	30,X	
01A0 B6 01 04		LDA A	SECTOR	GET SECTOR
01A3 A7 1F		STA A	31,X	
01A5 BD 70 06		JSR	FMS	CALL FMS

DUMP FILE UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 3

01A8 26 01		BNE	READ2	ERRORS?
01AA 39		RTS		
01AB A6 01	READ2	LDA A	1,X	GET ERROR TYPE
01AD 81 03		CMP A	#3	FILE EXISTS?
01AF 27 09		BEQ	ERROR2	
01B1 81 04		CMP A	#4	NO FILE?
01B3 27 0A		BEQ	ERROR4	
01B5 BD 71 3C		JSR	RPTERR	REPORT ERROR
01B8 20 10		BRA	ERROR8	
01BA CE 01 EF	ERROR2	LDX	#FEST	POINT TO STRING
01BD 20 08		BRA	ERROR7	
01BF CE 01 D0	ERROR4	LDX	#NSST	POINT TO STRING
01C2 20 03		BRA	ERROR7	
01C4 CE 01 DD	ERROR6	LDX	#ILST	POINT TO STRING
01C7 BD 71 18	ERROR7	JSR	PSTRNG	PRINT STRING
01CA BD 78 03	ERROR8	JSR	FMSCLS	CLOSE, ALL
01CD 7E 71 03		JMP	WARMS	RETURN TO FLEX

* STRINGS

01D0 4E	NSST	FCC	'NO SUCH FILE'
01DC 04		FCB	4
01DD 49	ILST	FCC	'ILLEGAL FILE NAME'
01EE 04		FCB	4
01EF 46	FEST	FCC	'FILE EXISTS'
01FA 04		FCB	4
		END	DUMP

NO ERROR(S) DETECTED

SYMBOL TABLE:

DUMP	0100	DUMP1	0108	DUMP11	0113	DUMP12	0116	DUMP2	0151
DUMP25	0156	DUMP27	0164	DUMP28	016E	DUMP6	01SD	ERROR2	01BA
ERROR4	01BF	ERROR6	01C4	ERROR7	01C7	ERROR8	01CA	FCB	7740
FEST	01EF	FMS	7806	FMSCLS	7803	GETCHR	710F	GETFIL	7127
ILST	01DD	LOOP	0134	NSST	01D0	OUTH	0190	OUTHEX	7139
PASS	0105	PCRLF	711E	PSTRNG	7118	PUTCHR	7112	READ	0198
READ2	01AB	RPTERR	713C	SECTOR	0104	SETEXT	712D	TEMP	0106
TRACK	0103	YN	0102	WARMS	7103				

OLOAD

The OLOAD command is used to load a binary file into memory with a specified offset address. No code is modified during the load. This utility is useful for PROM programming applications.

DESCRIPTION

The general syntax of the OLOAD command is:

```
OLOAD,<file spec>[,<offset>]
```

where the file spec defaults to a BIN extension and to the working drive. The optional offset is a hex value which is to be added to the normal load address. If the offset is not specified, it is assumed zero which causes OLOAD to act exactly like the GET command. The offset addition will wrap around the end address of \$FFFF. As an example, if a file normally loaded at location \$6000, and an offset of \$A000 was specified, the file would be loaded at \$0000. An example of the OLOAD command follows:

```
+++OLOAD,XDATA,2000
```

This would cause the file named XDATA.BIN to be loaded into memory offset by \$2000 from its normal load address. If XDATA normally resided at \$0100, the new location would be \$2100. Remember that no code is modified, so unless the binary file is relocatable code, it will not run in its new location.

```

* OFFSET LOAD UTILITY
*
*
* COPYRIGHT (C) 1978 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*

```

7600		ORG	\$7600	\$A10~
7127	GETFIL	EQU	\$7127	\$AD2D
712D	SETEXT	EQU	\$712D	\$AD33
713F	GETHEX	EQU	\$713F	\$AD42
789C	OFFSET	EQU	\$709C	\$AC1B
7740	FCB	EQU	\$7740	\$AD4C
7103	WARMS	EQU	\$7103	\$ADCB
7806	FMS	EQU	\$7806	\$B406
712A	DLOAD	EQU	\$712A	\$AD5C
7118	PSTRNG	EQU	\$7118	\$AD1E
713C	RPTERR	EQU	\$713C	\$AD3F
7803	FMSCLS	EQU	\$7803	\$B4C3

* PROGRAM STARTS HERE

7600 20 01	LOAD	BRA	LOAD1	
7602 01	VN	FCB	1	VERSION NUMBER
7603 CE 77 40	LOAD1	LDX	#FCB	POINT TO FCB
7606 BD 71 27		JSR	GETFIL	GET FILE SPEC
7609 25 46		BCS	ERROR6	ERROR?
760B 4F		CLR A		SET BINARY EXTENSION
760C CE 77 40		LDX	#FCB	POINT TO FCB
760F BD 71 2D		JSR	SETTEXT	
7612 BD 71 3F		JSR	GETHEX	GET OFFSET
7615 25 19		BCS	LOAD6	ERROR?
7617 FF 70 9C		STX	OFFSET	SET OFFSET
761A CE 77 40		LDX	#FCB	POINT TO FCB
761D 86 01		LDA A	#1	OPEN FILE FOR READ
761F A7 00		STA A	0,X	
7621 BD 78 06		JSR	FMS	
7624 26 12		BNE	LOAD7	ERRORS?
7626 86 FF		LDA A	#\$FF	SET FOR BINARY
7628 A7 3B		STA A	59,X	
762A BD 71 2A		JSR	DLJAD	DO LOAD
762D 7E 71 03	LOAD3	JMP	WARMS	RETURN TO FLEX
7630 CE 76 5D	LOAD6	LDX	#JI NUM	POINT TO STRING

OFFSET LOADER

TSC MNEMONIC ASSEMBLER

PAGE 2

7633 BD 71 18		JSR	PSTRNG	PRINT STRING
7636 20 F5		BRA	LOAD3	
7638 A6 01	LOAD7	LDA A	1,X	GET ERROR TYPE
763A 81 03		CMP A	#3	FILE EXISTS?
763C 27 09		BEQ	ERROR2	
763E 81 04		CMP A	#4	NO FILE?
7640 27 0A		BEQ	ERROR4	
7642 BD 71 3C		JSR	RPTERR	REPORT ERROR
7645 20 10		BRA	ERROR8	
7647 CE 76 BB	ERROR2	LDX	#FEST	POINT TO STRING
764A 20 08		BRA	ERROR7	
764C CE 76 6C	ERROR4	LDX	##NSST	POINT TO STRING
764F 20 03		BRA	ERROR7	
7651 CE 76 79	ERROR6	LDX	#ILST	POINT TO STRING
7654 BD 71 18	ERROR7	JSR	PSTRNG	PRINT STRING
7657 BD 72 03	ERROR8	JSR	FMSCLS	CLOSE ALL
765A 2E 71 03		JMP	WARM	

* STRINGS

765D 49	ILNUM	FCC	'ILLEGAL OFFSET'
766B 04		FCB	4
766C 4E	NSST	FCC	'NO SUCH FILE'
7678 04		FCB	4
7679 49	ILST	FCC	'ILLEGAL FILE NAME'
768A 04		FCB	4
768B 46	FEST	FCC	'FILE EXISTS'
7696 04		FCB	4
		END	1 LOAD

NO ERROR(S) DETECTED

SYMBOL TABLE:

CHECK

The CHECK utility is used to compare two disk files. The result of the comparison will be reported to the terminal.

DESCRIPTION

The general syntax of the CHECK command is:

```
CHECK,<file spec 1>,<file spec 2>
```

where the file specs default to a TXT extension and to the working drive. File one will be read and compared against file two one character at a time. The files may be text or binary type files. The result of the comparison will be reported to the terminal (files are identical or not). An example follows:

```
+++CHECK,REPORT1,REPORT2
```

This command line would cause the file named REPORT1.TXT on the working drive to be compared to the file named REPORT2.TXT.

*
 * CHECK FILES UTILITY
 *
 * COPYRIGHT (C) 1978 BY
 * TECHNICAL SYSTEMS CONSULTANTS, INC.
 * BOX 2574
 * WEST LAFAYETTE, INDIANA 47906
 *

* EQUATES

7806	FMS	EQU	\$7806	\$A406
7803	FMSCLS	EQU	\$7803	\$A403
7103	WARMs	EQU	\$7103	\$A003
710F	GETCHR	EQU	\$710F	\$A015
7112	PUTCHR	EQU	\$7112	\$A01D
7118	PSTRNG	EQU	\$7118	\$A01E
711E	PCRLF	EQU	\$711E	\$A024
7127	GETFIL	EQU	\$7127	\$A02D
712D	SETEXT	EQU	\$712D	\$A033
713C	RPTERR	EQU	\$713C	\$A03F
7740	FCB	EQU	\$7740	\$A040

* PROGRAM STARTS HERE

0100		ORG	\$0100	\$A100
0100 20 01	CHECK	BRA	CHECK1	
0102 01	VN	FCB	1	VERSION NUMBER
0103 CE 77 40	CHECK1	LDX	#FCB	POINT TO FCB
0106 BD 71 27		JSR	GETFIL	GET FILE NAME
0109 25 49		BCS	ERROR6	ERROR?
010B 86 01		LDA A	#1	
010D A7 00		STA A	0,X	
010F BD 71 2D		JSR	SETEXT	SET DEFAULT EXT
0112 CE 77 40		LDX	#FCB	
0115 BD 78 06		JSR	FMS	OPEN FILE
0118 26 21		BNE	CHECK2	ERRORS?
011A 86 FF		LDA A	#\$FF	SET FOR BINARY
011C A7 3B		STA A	59,X	
011E CE 02 05		LDX	#FCB2	POINT TO FCB
0121 BD 71 27		JSR	GETFIL	GET NEXT NAME
0124 25 2E		BCS	ERROR6	ANY ERRORS
0126 86 01		LDA A	#1	
0128 A7 00		STA A	0,X	SET FOR READ
012A BD 71 2D		JSR	SETEXT	SET TXT EXTENSION
012D CE 02 05		LDX	#FCB2	
0130 BD 78 06		JSR	FMS	OPEN FILE
0133 26 06		BNE	CHECK2	
0135 86 FF		LDA A	#\$FF	SET FOR BINARY

0137 A7 3B		STA R	59, X	
0139 20 27		BRA	CHECK5	
013B A6 01	CHECK2	LDA A	1, X	GET ERROR TYPE
013D 81 03		CMP A	#3	FILE EXISTS?
013F 27 09		BEQ	ERROR2	
0141 81 04		CMP A	#4	NO FILE?
0143 27 0A		BEQ	ERROR4	
0145 BD 71 3C		JSR	RPTERR	REPORT ERROR
0148 20 10		BRA	ERROR8	
014A CE 01 F9	ERROR2	LDX	#FEST	POINT TO STRING
014D 20 08		BRA	ERROR7	
014F CE 01 DA	ERROR4	LDX	#NSST	POINT TO STRING
0152 20 03		BRA	ERROR7	
0154 CE 01 E7	ERROR6	LDX	#ILST	POINT TO STRING
0157 BD 71 18	ERROR7	JSR	PSTRNG	PRINT STRING
015A BD 78 03	ERROR8	JSR	FMSCLS	CLOSE ALL
015D 7E 71 03	CHECK3	JMP	WARMs	RETURN TO DOS
0160 20 D9		BRA	CHECK2	REPORT ERROR
0162 CE 77 40	CHECK5	LDX	#FCB	POINT TO FCB
0165 BD 78 06		JSR	FMS	GET A CHARACTER
0168 26 14		BNE	CHECK6	ERRORS?
016A 16		TAB		SAVE IT
016B CE 02 05		LDX	#FCB2	GET FROM 2ND FILE
016E BD 78 06		JSR	FMS	
0171 26 C9		BNE	CHECK2	ERRORS?
0173 11		CBA		COMPARE CHARACTERS
0174 27 EC		BEQ	CHECK5	
0176 CE 01 9B	CHEC55	LDX	#NCST	ERROR STRING
0179 BD 71 18		JSR	PSTRNG	PRINT IT
017C 20 DF		BRA	CHECK3	
017E A6 01	CHECK6	LDA R	1, X	GET ERROR #
0180 81 08		CMP A	#8	IS IT EOF
0182 26 B7		BNE	CHECK2	
0184 CE 02 05		LDX	#FCB2	CHECK FILE 2
0187 BD 78 06		JSR	FMS	GET CHARACTER
018A 27 EA		BEQ	CHEC55	EOF?
018C A6 01		LDA R	1, X	GET ERROR #
018E 81 08		CMP A	#8	
0190 26 A9		BNE	CHECK2	
0192 CE 01 BA		LDX	#CMST	POINT TO STRING
0195 BD 71 18		JSR	PSTRNG	PRINT IT
0198 7E 71 03		JMP	WARMs	RETURN TO FLEX

* STRINGS

019B 54	NCST	FCC	'THE FILES CHECKED DO NOT MATCH'
01B9 04		FCB	4
018A 54	CMST	FCC	'THE FILES CHECKED ARE IDENTICAL'
01D9 04		FCB	4
01DA 4E	NSST	FCC	'NO SUCH FILE'
01E6 04		FCB	4
01E7 49	ILST	FCC	'ILLEGAL FILE NAME'
01F8 04		FCB	4
01F9 46	FEST	FCC	'FILE EXISTS'

0204 04		FCB	4	
0205	FCB2	EQU	*	SECOND FCB
		END	CHECK	

NO ERROR(S) DETECTED

SYMBOL TABLE:

CHEC55 0176	CHECK 0100	CHECK1 0103	CHECK2 013B	CHECK3 015D
CHECK5 0162	CHECK6 017E	CMST 01BA	ERROR2 014A	ERROR4 014F
ERROR6 0154	ERROR7 0157	ERROR8 015A	FCB 7740	FCB2 0205
FEST 01F9	FMS 7806	FMSCLS 7803	GETCHR 710F	GETFIL 7127
ILST 01E7	NCST 019B	NSST 01DA	PCRLF 711E	PSTRNG 7118
PUTCHR 7112	RPTERR 713C	SETEXT 712D	VN 0102	WARMS 7103

CMPMEM

The CMPMEM command compares the contents of a binary file on the disk to the contents of memory where it should be loaded. This is useful for program debugging and memory problem detection.

DESCRIPTION

The general syntax of the CMPMEM command is:

```
CMPMEM,<file spec>
```

where the file spec defaults to a BIN extension and to the working drive. The file specified will be read just as if it were to be loaded into memory, but instead, each byte will be compared to what already exists in memory. If any differences are found, they will be printed out as the address, followed by the data in memory at that location, followed by the data from the disk file. All differences will be printed on the output device. An example follows:

```
+++CMPMEM,FENCE
```

This would cause the file named FENCE.BIN on the working drive to be read and compared to the actual memory contents throughout the load address range of the file.

*
 * COMPARE BINARY FILE TO MEMORY
 *
 *
 * COPYRIGHT (C) 1978 BY
 *
 * TECHNICAL SYSTEMS CONSULTANTS, INC.
 * P. O. BOX 2574
 * WEST LAFAYETTE, INDIANA 47906
 *
 *

0100 ORG \$0100 *✓A100*

* EQUATES

7103	WARMS	EQU	\$7103	<i>✓ADC3</i>
7806	FMS	EQU	\$7806	<i>✓134C6</i>
7803	FMSCLS	EQU	\$7803	<i>✓134C3</i>
7740	FCB	EQU	\$7740	<i>✓A04C</i>
7127	GETFIL	EQU	\$7127	<i>✓AD2D</i>
712D	SETEXT	EQU	\$712D	<i>✓AD33</i>
713C	RPTERR	EQU	\$713C	<i>✓AD3F</i>
7118	PSTRNG	EQU	\$7118	<i>✓AD1E</i>
711E	PCRLF	EQU	\$711E	<i>✓AD24</i>
7112	PUTCHR	EQU	\$7112	<i>✓AD18</i>
710F	GETCHR	EQU	\$710F	<i>✓AD15</i>
7139	OUTHEX	EQU	\$7139	<i>✓AD3C</i>

0100 20 04	CMP	BRA	CMP1	
0102 01	VN	FCB	1	VERSION NUMBER
0103 00 00	ADDR	FDB	0	
0105 00	TEMP	FCB	0	
0106 CE 77 40	CMP1	LDX	#FCB	POINT TO FCB
0109 BD 71 27		JSR	GETFIL	GET FILE NAME
010C 25 57		BCS	ERROR6	ERRORS?
010E 4F		CLR A		
010F BD 71 2D		JSR	SETEXT	DEFAULT TO BIN
0112 CE 77 40		LDX	#FCB	
0115 86 01		LDA A	#1	OPEN FOR READ
0117 A7 00		STA A	0,X	
0119 BD 78 06		JSR	FMS	CALL FMS
011C 26 2E		BNE	CNP6	ERRORS?
011E 86 FF		LDA A	#\$FF	SET FOR BINARY
0120 A7 3B		STA A	59,X	SET TO BINARY
0122 BD 01 71	CMP2	JSR	READ	GET A CHARACTER
0125 81 02		CMP A	#2	IS IT BOT ?
0127 26 F9		BNE	CMP2	
0129 8D 46		BSR	RIAD	GET ADDRESS

012B 36		PSH A		
012C 8D 43		BSR READ		GET CHARACTER
012E 33		PUL B		
012F F7 01 03		STA B ADDR		SAVE ADDRESS
0132 B7 01 04		STA A ADDR+1		
0135 8D 3A		BSR READ		GET COUNT
0137 16		TAB		SAVE COUNT
0138 27 E8		BEQ CMP2		
013A 8D 35	CMP4	BSR READ		GET CHARACTER
013C FE 01 03		LDX ADDR		GET ADDRESS
013F A1 00		CMP A 0, X		
0141 26 3F		BNE NOCMP		CHECK IF SAME
0143 08	CMP5	INX		
0144 FF 01 03		STX ADDR		SAVE NEW ADDRESS
0147 5A		DEC B		DEC THE COUNT
0148 26 F0		BNE CMP4		
014A 20 D6		BRA CMP2		
014C A6 01	CMP6	LDA A 1, X		GET ERROR TYPE
014E 81 03		CMP A #3		FILE EXISTS?
0150 27 09		BEQ ERROR2		
0152 81 04		CMP A #4		NO FILE?
0154 27 0A		BEQ ERROR4		
0156 BD 71 3C		JSR RPTERR		REPORT ERROR
0159 20 10		BRA ERROR8		
015B CE 01 CE	ERROR2	LDX #FEST		POINT TO STRING
015E 20 08		BRA ERROR7		
0160 CE 01 AF	ERROR4	LDX #NSST		POINT TO STRING
0163 20 03		BRA ERROR7		
0165 CE 01 BC	ERROR6	LDX #ILST		POINT TO STRING
0168 BD 71 18	ERROR7	JSR PSTRNG		PRINT STRING
016B BD 78 03	ERROR8	JSR FM3CLS		CLOSE ALL
016E 7E 71 03	CMP8	JMP WARMS		RETURN TO FLEX

* READ NEXT CHARACTER

0171 CE 77 40	READ	LDX #FCB		POINT TO FCB
0174 BD 78 06		JSR FMS		GET CHARACTER
0177 27 08		BEQ READ4		ERRORS?
0179 A6 01		LDA A 1, X		GET ERROR NUM
017B 81 08		CMP A #8		IS IT EOF?
017D 27 EF		BEQ CMP8		
017F 20 CB		BRA CMP6		
0181 39	READ4	RTS		RETURN

* BYTE DOES NOT COMPARE

0182 B7 01 05	NOCMP	STA A TEMP		SAVE CHARACTER
0185 37		PSH B		
0186 BD 71 1E		JSR PCRLF		OUTPUT CR & LF
0189 CE 01 03		LDX #ADDR		POINT TO ADDRESS
018C BD 71 39		JSR OUTHEX		OUTPUT ADDRESS
018F 42		INX		
0190 BD 71 39		JSR OUTHEX		
0193 8C 20		LDH A #120		

MEMORY COMPARE

TSC MNEMONIC ASSEMBLER

PAGE 3

0195 BD 71 12	JSR	PUTCHR	OUTPUT SPACE
0198 FE 01 03	LDX	ADDR	PRINT DATA
019B BD 71 39	JSR	OUTHEX	
019E S6 20	LDA R	##\$20	
01A0 BD 71 12	JSR	PUTCHR	OUTPUT SPACE
01A3 CE 01 05	LDX	#TEMP	
01A6 BD 71 39	JSR	OUTHEX	
01A9 33	PUL B		
01AA FE 01 03	LDX	ADDR	RESTORE POINTER
01AD 20 94	BRA	CMP5	

* STRINGS

01AF 4E	NSST	FCC	'NO SUCH FILE'
01BB 84		FCB	4
01BC 49	ILST	FCC	'ILLEGAL FILE NAME'
01CD 84		FCB	4
01CE 46	FEST	FCC	'FILE EXISTS'
01D9 84		FCB	4
	END	CMP	

NO ERROR(S) DETECTED

SYMBOL TABLE:

ADDR	0183	CMP	0100	CMP1	0106	CMP2	0122	CMP4	013A
CMP5	0143	CMP6	014C	CMP8	016E	ERROR2	015B	ERROR4	0160
ERROR6	0165	ERROR7	0168	ERROR8	016B	FCB	7740	FEST	01CE
FMS	7806	FMSCLS	7803	GETCHR	710F	GETFIL	7127	ILST	01BC
NOCMP	0182	NSST	01AF	OUTHEX	7139	PCRLF	711E	PSTRNG	7118
PUTCHR	7112	READ	0171	READ4	0181	RPTERR	713C	SETTEXT	712D
TEMP	0105	VN	0102	WARMS	7103				

FILTYP

The FILTYP command is used to determine the type of a file, either binary or text. This is useful when non-standard extensions have been used and the file type has been forgotten.

DESCRIPTION

The general syntax of the FILTYP command is:

```
FILTYP,<file spec>
```

where the file defaults to the working drive. Upon executing this command, the system will report the file to be either a TEXT type file, or a BINARY type file (a file which may be loaded into memory). An example will demonstrate its use.

```
+++FILTYP,MYSTERY.XYZ
```

The system will report the type of the file named MYSTERY.XYZ found on the working drive.

```

* DETERMINE FILE TYPE
*
*
* COPYRIGHT (C) 1978 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*

```

7600

ORG \$7600

\$A1000

* EQUATES

7103	WARMIS	EQU	\$7103	\$A1003
7740	FCB	EQU	\$7740	\$A1840
7127	GETFIL	EQU	\$7127	\$A1D2D
712D	SETEXT	EQU	\$712D	\$A1D33
7806	FMS	EQU	\$7806	\$A1B406
713C	RPTERR	EQU	\$713C	\$A1D3F
7118	PSTRNG	EQU	\$7118	\$A1D1E
7803	FMSCLS	EQU	\$7803	\$A1B403

* PROGRAM STARTS HERE

7600 20 01

TYP BRA TYP1

7602 01

VN FCB 1 VERSION NUMBER

7603 CE 77 40	TYP1	LDX #FCB	POINT TO FCB
7606 BD 71 27		JSR GETFIL	GET FILE NAME
7609 25 3D		BCS ERROR6	ERROR?
760B 86 01		LDA A #1	SET FOR READ
760D A7 00		STA A 0,X	
760F BD 71 2D		JSR SETEXT	SET EXTENSION
7612 CE 77 40		LDX #FCB	
7615 BD 78 06		JSR FMS	OPEN FILE
7618 26 15		BNE TYP6	
761A BD 78 06		JSR FMS	READ CHARACTER
761D 81 02		CMP A #2	IS IT BINARY?
761F 27 05		BEQ TYP4	
7621 CE 76 54		LDX #TXTST	POINT TO STRING
7624 20 03		BRA TYP5	
7626 CE 76 68	TYP4	LDX #BINST	POINT TO STRING
7629 BD 71 18	TYP5	JSR PSTRNG	PRINT IT
762C 7E 71 03		JMP WARMIS	RETURN TO FLEX
762F A6 01		LDA A 1,X	GET ERROR TYPE
7631 81 03		CMP A #3	FILE EXISTS?
7633 27 09		BEQ ERROR2	
7635 81 04		CMP A #4	NO FILE?

7637 27 0A		BEQ	ERROR4	
7639 BD 71 3C		JSR	RPTERR	REPORT ERROR
763C 20 10		BRA	ERROR8	
763E CE 76 9D	ERROR2	LDX	#FEST	POINT TO STRING
7641 20 08		BRA	ERROR7	
7643 CE 76 7E	ERROR4	LDX	#NSST	POINT TO STRING
7646 20 03		BRA	ERROR7	
7648 CE 76 8B	ERROR6	LDX	#ILST	POINT TO STRING
764B BD 71 18	ERROR7	JSR	PSTRNG	PRINT STRING
764E BD 78 03	ERROR8	JSR	FMSCLS	CLOSE ALL
7651 7E 71 03		JMP	WARMS	

* STRINGS *

7654 46	TXTST	FCC	'FILE IS A TEXT FILE'
7667 04		FCB	4
7668 46	BINST	FCC	'FILE IS A BINARY FILE'
7670 04		FCB	4
767E 4E	NSST	FCC	'NO SUCH FILE'
768A 04		FCB	4
768B 49	ILST	FCC	'ILLEGAL FILE NAME'
769C 04		FCB	4
769D 46	FEST	FCC	'FILE EXISTS'
76A8 04		FCB	4
	END	TYP	

NO ERROR(S) DETECTED

SYMBOL TABLE:

BINST	7668	ERROR2	763E	ERROR4	7643	ERROR6	7648	ERROR7	764B
ERROR8	764E	FCB	7740	FEST	769D	FMS	7806	FMSCLS	7803
GETFIL	7127	ILST	768B	NSST	767E	PSTRNG	7118	RPTERR	713C
SETEXT	712D	TXTST	7654	TYP	7600	TYP1	7603	TYP4	7626
TYP5	7629	TYP6	762F	YN	7602	WARMS	7103		

DUP

The DUP command is used to list the file names contained in one disk's directory which are not duplicated in a second disk's directory. This is a useful utility for comparing files on original disks to those on a backup diskette.

DESCRIPTION

The general syntax of the DUP command is:

DUP,<drive number>,<drive number>

where the disk's directory in the first drive number specified is to be compared to the second. The files which exist on the first drive but not on the second will be listed on the output device. As an example:

+++DUP,0,1

This would cause all file names which were on the diskette in drive zero but not on drive one to be listed on the output device.

```

*
* NON- DUPLICATE FILE CHECK
*
*
* COPYRIGHT (C) 1978 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*
* EQUATES

```

713F	GETHEX	EQU	\$713F	\$AD42
7133	OUTDEC	EQU	\$7133	\$AD34
7103	WARMS	EQU	\$7103	\$A003
713C	RPTERR	EQU	\$713C	\$A13F
7806	FMS	EQU	\$7806	\$B4C6
711E	PCRLF	EQU	\$711E	\$A124
7112	PUTCHR	EQU	\$7112	\$A110
7740	FCB	EQU	\$7740	\$A040
7803	FMSCLS	EQU	\$7803	\$B4C3
7118	PSTRNG	EQU	\$7118	\$AD14

* DUP STARTS HERE

0100		ORG	\$0100	\$A1C0
0100 20 09	DUP	BRA	DUP1	
0102 01	VN	FCB	1	
0103 00 00	DRV1	FDB	0	
0105 00 00	DRV2	FDB	0	
0107 00 00	TEMP	FDB	0	
0109 00 00	TEMP2	FDB	0	
010B BD 71 3F	DUP1	JSR	GETHEX	GET DRIVE
010E 25 13		BCS	DUP28	
0110 FF 01 03		STX	DRV1	
0113 BD 71 3F		JSR	GETHEX	GET NEXT DRIVE
0116 25 08	DUP2	BCS	DUP28	ERROR?
0118 FF 01 05		STX	DRV2	SAVE IT
011B B6 01 04		LDA A	DRV1+1	CHECK EQUAL DRIVES
011E B1 01 06		CMP A	DRV2+1	
0121 26 03		BNE	DUP3	
0123 7E 01 B5	DUP28	JMP	DUP8	
0126 CE 02 31	DUP3	LDX	#FCB2	POINT TO FCB
0129 B6 01 06		LDA A	DRV2+1	SET DRIVE
012C A7 03		STA A	3,X	
012E CE 77 40		LDX	#FCB	POINT TO FCB
0131 B6 01 04		LDA A	DRV1+1	GET DRIVE
0134 A7 03		STA A	3,X	SET IT

0136 86 06	LDA R	#6	OPEN DIR	
0138 A7 00	STA R	0,X		
013A BD 78 06	JSR	FMS	CALL FMS	
013D 26 57	BNE	DUP?	ERROR?	
013F CE 77 40	DUP4	LDX	#FCB	POINT TO DIR
0142 86 07		LDA R	#7	GET IR FROM DRIVE
0144 A7 00		STA R	0,X	
0146 BD 78 06		JSR	FMS	CALL FMS
0149 26 4B		BNE	DUP?	ERROR?
014B 6D 04		TST	4,X	CHECK ENTRY
014D 2B F0		BMI	DUP4	EMPTY?
014F 27 64		BEQ	DUP8	FINISHED?
0151 C6 0B		LDA B	#11	SET COUNT
0153 CE 77 40		LDX	#FCB	SET POINTERS
0156 FF 01 07		STX	TEMP	
0159 CE 02 31		LDX	#FCB2	
015C FF 01 09		STX	TEMP2	SAVE THEM
015F FE 01 07	DUP5	LDX	TEMP	GET POINTER
0162 A6 04		LDA R	4,X	GET CHARACTER
0164 08		INX		BUMP THE POINTER
0165 FF 01 07		STX	TEMP	
0168 FE 01 09		LDX	TEMP2	GET 2ND POINTER
016B A7 04		STA R	4,X	PUT CHARACTER
016D 08		INX		BUMP THE POINTER
016E FF 01 09		STX	TEMP2	SAVE IT
0171 5A		DEC B		DEC THE COUNT
0172 26 EB		BNE	DUP5	REPRT?
0174 CE 02 31		LDX	#FCB2	POINT TO FCB
0177 86 01		LDA R	#1	OPEN FOR READ
0179 A7 00		STA R	0,X	
017B BD 78 06		JSR	FMS	CALL FMS
017E 26 0B		BNE	DUP6	ERROR?
0180 86 04		LDA R	#4	CLOSE FILE
0182 A7 00		STA R	0,X	
0184 BD 78 06		JSR	FMS	
0187 26 0D		BNE	DUP?	ERROR?
0189 20 B4		BRA	DUP4	REPEAT
018B A6 01	DUP6	LDA R	1,X	GET ERROR NUM
018D 81 04		CMP A	#4	IS IT NO FILE ERR?
018F 26 05		BNE	DUP?	
0191 BD 01 BB		JSR	PRNAM	PRINT NAME
0194 20 A9		BRA	DUP4	REPRT
0196 A6 01	DUP7	LDA R	1,X	GET ERROR TYPE
0198 81 03		CMP A	#3	FILE EXISTS?
019A 27 09		BEQ	ERROR2	
019C 81 04		CMP A	#4	NO FILE?
019E 27 0A		BEQ	ERROR4	
01A0 BD 71 3C		JSR	RPTERR	REPORT ERROR
01A3 20 10		BRA	DUP8	
01A5 CE 02 25	ERROR2	LDX	#FEST	POINT TO STRING
01A8 20 08		BRA	ERROR?	
01AA CE 02 06	ERROR4	LDX	#INST	POINT TO STRING
01AD 20 03		BRA	ERROR?	
01AF CE 02 13	ERROR6	LDX	#ILST	POINT TO STRING

01B2 BD 71 18	ERROR7	JSR	PSTRNG	PRINT STRING
01B5 BD 71 1E	DUP8	JSR	PCRLF	PRINT CR & LF
01B8 7E 71 03		JMP	WARM8	

* PRINT FILE NAME

01BB BD 71 1E	PRNAM	JSR	PCRLF	PRINT CR & LF
01BE CE 01 03		LDX	#DRV1	POINT TO DRIVE
01C1 C6 01		LDR B	#1	
01C3 BD 71 33		JSR	OUTDEC	OUTPUT IT
01C6 86 2E		LDA A	#1.	PRINT 1.
01C8 BD 71 12		JSR	PUTCHR	
01CB CE 02 55		LDX	#FCB2+36	
01CE C6 08		LDA B	#8	SET COUNT
01D0 8D 18		BSR	PRNT	PRINT NAME
01D2 86 2E		LDA A	#1.	PRINT 1.
01D4 BD 71 12		JSR	PUTCHR	
01D7 C6 03		LDA B	#3	SET CONT
01D9 8D 0F		BSR	PRNT	PRINT EXT
01DB CE 01 F8		LDX	#STR	POINT TO STRING
01DE C6 0E		LDA B	#14	SET COUNT
01E0 8D 08		BSR	PRNT	PRINT IT
01E2 CE 01 05		LDX	#DRV2	POINT TO DRIVE
01E5 5F		CLR B		
01E6 BD 71 33		JSR	OUTDEC	PRINT IT
01E9 39		RTS		RETURN

* PRINT STRING

01EA A6 00	PRNT	LDA A	0,X	GET CHARACTER
01EC 26 02		BNE	PRNT2	
01EE 86 20		LDA A	#\$20	SETUP SPACE
01F0 BD 71 12	PRNT2	JSR	PUTCHR	PRINT CHAR
01F3 08		INX		
01F4 5A		DEC B		DEC THE COUNT
01F5 26 F3		BNE	PRNT	
01F7 39		RTS		RETURN

* STRINGS

01F8 20	STR	FCC	' NOT ON DRIVE '	
0206 4E	NSST	FCC	'NO SUCH FILE'	
0212 04		FCB	4	
0213 49	ILST	FCC	'ILLEGAL FILE NAME'	
0224 04		FCB	4	
0225 46	FEST	FCC	'FILE EXISTS'	
0230 04		FCB	4	
0231	FCB2	RMB	192 32.0	
		END	DUP	

MAP

The MAP utility is used for determining the load addresses and transfer address of a binary file. This command is useful in conjunction with the SAVE command.

DESCRIPTION

The general syntax of the MAP command is:

```
MAP,<file spec>
```

where the file spec defaults to a BIN extension and to the working drive. The beginning and ending addresses of each block of object code will be printed on the terminal. If a transfer address is contained in the file, it will be printed at the end of the list of addresses. If more than one transfer address is found in a file, only the effective one (the last one encountered) will be displayed. An example will demonstrate the use of MAP.

```
+++MAP,MONITOR
```

This command line would cause the load addresses and transfer address (if one exists) of the file named MONITOR.BIN to be displayed at the terminal.

```

*
* MAP UTILITY
*
* COPYRIGHT (C) 1978 BY
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* BOX 2574, W. LAFAYETTE, IN 47906
*
* GIVES STARTING AND ENDING ADDRESSES OF DATA
* BLOCKS AND TRANSFER ADDRESS FOR BINARY FILES.
* SAMPLE COMMAND: +++INFO,FILENAME
*
* DOS EQUATES
*

```

7103	WARM\$	EQU	\$7103	\$/A D C 3
7127	GETFIL	EQU	\$7127	\$/A D 2 D
7118	PSTRNG	EQU	\$7118	\$/A D 1 E
711E	PCRLF	EQU	\$711E	\$/A M 2 4
7112	PUTCHR	EQU	\$7112	\$/A D 1 C
712D	SETEXT	EQU	\$712D	\$/A D 3 3
7130	ADDBX	EQU	\$7130	\$/A D 3 6
7139	OUTHEX	EQU	\$7139	\$/A D 3 C
713C	RPTERR	EQU	\$713C	\$/A M 3 F
7806	FMS	EQU	\$7806	\$/B 4 C 6
7803	FMSCLS	EQU	\$7803	\$/B 4 C 3
7740	FCB	EQU	\$7740	\$/A D 4 0
0100		ORG	\$0100	\$/A 1 C - C

* PROGRAM START

0100 20 07	INFO	BRA	INFO1	
0102 01	VN	FCB	1	VERSION NUMBER

* TEMPORARY STORAGE

0103	TEMP	RMB	2	
0105 FF FF	PREV	FDB	\$FFFF	
0107 FF FF	TRNSFR	FDB	\$FFFF	

* MAIN PROGRAM

0109 CE 77 40	INFO1	LDX	#FCB	
010C BD 71 27		JSR	GETFIL	GET FILE SPEC
010F 24 03		BCC	INFO2	
0111 7E 01 E8		JMP	INFO14	JUMP IF ILLEGAL
0114 CE 77 40	INFO2	LDX	#FCB	
0117 86 00		LDA A	#0	
0119 BD 71 2D		JSR	SETEXT	BIN DEFAULT EXTENSION
011C CE 77 40		LDX	#FCB	
011F 86 01		LDA A	#1	OPEN FILE FOR READ
0121 A7 00		STA A	0,X	
0123 BD 79 06		JSR	FMS	
0126 27 03		BEQ	INFO3	
0128 7E 01 D9		JMP	INFO12	JUMP IF ERROR

012B 86 FF		INFO03	LDA A #FF	SET NON-SPACE COMPRESS
012D A7 3B			STA A 59,X	
012F 8D 20			BSR FILECH	GET A FILE CHAR.
0131 81 02			CMP A #\$02	A BINARY FILE?
0133 27 39			BEQ INFO05	
0135 7E 01 ED			JMP INFO015	ERROR IF NOT
0138 CE 77 40	GETADR		LDX #FCB	GET M. S. BYTE
013B BD 78 06			JSR FMS	
013E 26 0C			BNE GETAD1	
0140 B7 01 03			STA A TEMP	
0143 BD 79 06			JSR FMS	GET L. S. BYTE
0146 26 04			BNE GETAD1	
0148 B7 01 04			STA A TEMP+1	STORE ADDRESS AT TEMP
014B 39			RTS	
014C 32	GETAD1		PUL A	FIX STACK
014D 32			PUL A	
014E 7E 01 A7			JMP INFO08	GO HANDLE ERROR
0151 CE 77 40	FILECH		LDX #FCB	GET A CHAR FROM FILE
0154 BD 78 06			JSR FMS	
0157 26 F3			BNE GETAD1	EXIT IF AN ERROR
0159 39			RTS	
015A 8D F5	INFO04		BSR FILECH	GET A CHAR
015C 81 02			CMP A #\$02	RECORD ID?
015E 27 0E			BEQ INFO05	
0160 81 16			CMP A #\$16	TRANSFER ID?
0162 26 F6			BNE INFO04	LOOP IF NEITHER
0164 8D D2			BSR GETADR	GET TRANSFER ADDRESS
0166 FE 01 03			LDX TEMP	
0169 FF 01 07			STX TRNSFR	SAVE IT
016C 20 EC			BRA INFO04	
016E 8D C8	INFO05		BSR GETADR	GET LOAD ADDRESS
0170 FE 01 05			LDX PREV	GET LAST ADDRESS
0173 8C FF FF			CPX #\$FFFF	ANY PRINTED YET?
0176 27 08			BEQ INFO06	SKIP IF NOT
0178 BC 01 03			CPX TEMP	DISCONTINUOUS CODE?
017B 27 15			BEQ INFO07	SKIP IF NOT
017D BD 01 F5			JSR PRTEND	ELSE, END THIS BLOCK
0180 BD 71 1E	INFO06		JSR PCRLF	
0183 CE 01 03			LDX #TEMP	PRINT START ADDRESS
0186 BD 71 39			JSR OUTHEX	
0189 08			INX	
018A BD 71 39			JSR OUTHEX	
018D 86 2D			LDA A #'-	OUTPUT A DASH
018F BD 71 12			JSR PUTCHR	
0192 8D BD	INFO07		BSR FILECH	GET DATA COUNT
0194 16			TAB	
0195 FE 01 03			LDX TEMP	
0198 37			PSH B	
0199 BD 71 30			JSR ADDBX	ADD TO RECORD ADDRESS
019C FF 01 05			STX PREV	STORE AWAY
019F 33			PUL B	
01A0 8D AF	INFO075		BSR FILECH	READ TO END OF RECORD
01A2 5A			DEC B	
01A3 26 FB			BNE INFO075	

01A5 20 B3		BRA	INFO4	REPEAT LOOP
01A7 A6 01	INFO8	LDA A	1,X	
01A9 81 08		CMP A	#8	EOF ERROR?
01AB 26 23		BNE	INFO10	
01AD 8D 46		BSR	PRTEND	PRINT LAST END ADDRESS
01AF BD 71 1E		JSR	PCRLF	
01B2 FE 01 07		LDX	TRNSFR	GET TRANSFER
01B5 8C FF FF		CPX	#\$FFFF	ANY FOUND?
01B8 27 0A		BEQ	INFO9	SKIP IF NOT
01BA CE 01 07		LDX	#TRNSFR	PRINT TRANSFER ADDRESS
01BD BD 71 39		JSR	OUTHEX	
01C0 08		INX		
01C1 BD 71 39		JSR	OUTHEX	
01C4 CE 77 40	INFO9	LDX	#FCB	CLOSE FILE
01C7 86 04		LDA A	#4	
01C9 A7 00		STA A	0,X	
01CB BD 78 06		JSR	FMS	
01CE 27 06		BEQ	INFO11	
01D0 BD 71 3C	INFO10	JSR	RPTERR	REPORT ERROR
01D3 BD 78 03		JSR	FMSCLS	
01D6 7E 71 03	INFO11	JMP	WARMs	BACK TO DOS
01D9 A6 01	INFO12	LDA A	1,X	
01DB 81 04		CMP A	#4	DOES FILE EXIST?
01DD 26 F1		BNE	INFO10	
01DF CE 02 02		LDX	#NOFST	IF NOT, REPORT IT
01E2 BD 71 18	INFO13	JSR	PSTRNG	
01E5 7E 71 03		JMP	WARMs	BACK TO DOS
01E8 CE 02 0F	INFO14	LDX	#ILLST	
01EB 20 F5		BRA	INFO13	
01ED CE 02 21	INFO15	LDX	#NOTBIN	
01F0 BD 71 18		JSR	PSTRNG	
01F3 20 CF		BRA	INFO9	
01F5 CE 01 05	PRTEND	LDX	#PREV	GET PREVIOUS ADDRESS
01F8 BD 71 39		JSR	OUTHEX	OUTPUT IT
01FB 08		INX		
01FC 6A 00		DEC	0,X	
01FE BD 71 39		JSR	OUTHEX	
0201 39		RTS		

* PRINT STRINGS

0202 4E	NOFST	FCC	'NO SUCH FILE'
020E 04		FCB	4
020F 49	ILLST	FCC	'ILLEGAL FILE NAME'
0220 04		FCB	4
0221 4E	NOTBIN	FCC	'NOT BINARY'
022B 04		FCB	4
	END		INFO

NO ERROR(S) DETECTED

DIR

The DIR utility is similar to the CAT command but displays all directory information associated with the file. This command gives a detailed look at the disk directory.

DESCRIPTION

The general syntax of the DIR command is:

```
DIR[,<drive list>][,<match list>]
```

where <drive list> and <match list> are the same as described in the CAT command. Each file name will be listed with its file number, starting disk address in hex (track-sector), ending disk address, and file size in number of sectors. On the larger FLEX systems, the file creation date and attributes will also be displayed. At the end of the DIR list, a disk file use summary is printed, giving the total number of files, the number of sectors used by those files, the remaining number of sectors (free sectors), and the size of the largest file found on the disk. The 'file number' associated with a file represents that file's location in the directory, so the file numbers may not be consecutive if a lot of files have been deleted from the disk. A few examples follow:

```
+++DIR  
+++DIR, I,A.T,FR
```

The first example would list all files on the working drive. The second example would list only those files on drive 1 whose names began with 'A' and extensions began with 'T', as well as those files whose names started with 'FR'.

*
 * "DIR" UTILITY PROGRAM
 *
 *
 * COPYRIGHT (C) 1978 BY
 *
 * TECHNICAL SYSTEMS CONSULTANTS, INC.
 * P. O. BOX 2574
 * WEST LAFAYETTE, INDIANA 47906
 *
 *

* GLOBAL VARIABLES

7082	EOL	EQU	\$7082	\$AC02
7740	FCB	EQU	\$7740	\$A040
708C	WASN	EQU	\$708C	\$AC0C
708C	ASN	EQU	WASN	
7091	LSTTRM	EQU	\$7091	\$AC11
7094	BUFPNT	EQU	\$7094	\$AC14
0020	SPC	EQU	\$20	
7803	FMSCLS	EQU	\$7803	\$B403
7806	FMS	EQU	\$7806	\$B406
7103	WARMs	EQU	\$7103	\$AD03
710F	GETCHR	EQU	\$710F	\$AD15
7112	PUTCHR	EQU	\$7112	\$AD10
7118	PSTRNG	EQU	\$7118	\$AD1E
711B	CLASS	EQU	\$711B	\$AD21
711E	PCRLF	EQU	\$711E	\$AD24
7121	NXTCH	EQU	\$7121	\$AD27
7127	GETFIL	EQU	\$7127	\$AD2D
7133	OUTDEC	EQU	\$7133	\$AD39
7139	OUTHEX	EQU	\$7139	\$AD3C
713C	RPTERR	EQU	\$713C	\$AD3F
713F	GETHEX	EQU	\$713F	\$AD42
0100		ORG	\$0100	\$A100
0100 20 22	DIR	BRA	DIR1	
0102 01	VN	FCB	1	VERSION NUMBER
0103 00	NAME	FCB	0, 0, 0, 0	
0107 00		FCB	0, 0, 0, 0	
0108 00	EXT	FCB	0, 0, 0	
010E 00	ALLFLG	FCB	0	
010F 00	LOOP	FCB	0	
0110 00 00	POINT	FDB	0	
0112 00 00	FNUM	FDB	0	
0114 00 00	FMAX	FDB	0	
0116 00 00	SECS	FDB	0	
0118 00 00	LARG	FDB	0	
011A 00 00	DEC1	FDB	0	
011C 00 00	TEMP	FDB	0	

011E 00 00	INDEX	FDB	0
0120 00 00	DATPNT	FDB	0
0122 00 00	DSNUM	FDB	0

* PROGRAM HERE

0124 FE 70 94	DIR1	LDX	BUFPNT	POINT TO BUFFER
0127 FF 01 10		STX	POINT	SAVE POS
012A BD 71 1E		JSR	PCRLF	PRINT CR & LF
012D FE 01 10	DIR22	LDX	POINT	GET POINTER
0130 A6 00		LDA A	0,X	GET CHARACTER
0132 81 2E		CMP A	#`.	IS IT PERIOD?
0134 27 09		BEQ	DIR25	
0136 BD 71 1B		JSP	CLASS	CLASSIFY IT
0139 25 04		BCS	DIR25	TERM?
013B 81 39		CMP A	#`9	IS IT NUMBER?
013D 23 12		BLS	DIR3	
013F 7D 01 0F	DIR25	TST	LOOP	LOOPING?
0142 27 03		BEQ	DIR29	
0144 7E 71 03	DIR27	JMP	WARM	RETURN TO FLEX
0147 CE 77 40	DIR29	LDX	#FCB	POINT TO FCB
014A B6 70 8C		LDA A	ASN	GET DRIVE
014D A7 03		STA A	3,X	SET IN FCB
014F 20 1E		BRA	DIR4	
0151 FE 01 10	DIR3	LDX	POINT	RESTORE POINTER
0154 FF 70 94		STX	BUFPNT	
0157 BD 71 21		JSR	NXTCH	GET NEXT CHAR
015A 84 03		AND A	#3	MASK NUMBER
015C CE 77 40		LDX	#FCB	POINT TO FCB
015F A7 03		STA A	3,X	SAVE IN DRIVE
0161 BD 71 21		JSR	NXTCH	GET NEXT CHAR
0164 25 03		BCS	DIR35	TERM?
0166 7E 02 2A		JMP	SERR	IF NOT, ERROR
0169 FE 70 94	DIR35	LDX	BUFPNT	SET POINTER
016C FF 01 10		STX	POINT	
016F CE 77 40	DIR4	LDX	#FCB	POINT TO FCB
0172 86 10		LDA A	#16	OPEN SYSTEM REC
0174 A7 00		STA A	0,X	
0176 BD 78 06		JSR	FMS	
0179 26 2A		BNE	DIR43	ERROR?
017B BD 03 2F		JSR	GETIR	GET FIRST RECORD
017E 26 25		BNE	DIR43	ERROR?
0180 CE 03 DB		LDX	#DIRST	POINT TO STRING
0183 BD 71 18		JSR	PSTRNG	PRINT IT
0186 CE 77 42		LDX	#FCB+2	POINT TO DRIVE
0189 6F 00		CLR	0,X	
018B 5F		CLR B		
018C BD 71 33		JSR	OUTDEC	PRINT IT
018F BD 71 1E		JSR	PCRLF	PRINT CR & LF
0192 CE 03 97		LDX	#HDR	POINT TO HEADER
0195 BD 71 18		JSR	PSTRNG	PRINT IT
0198 BD 71 1E		JSR	PCRLF	PRINT CR & LF
019B BD 02 38	DIR42	JSR	TSTTRM	TEST TERMINATOR
019E 26 08		BNE	DIP45	

01A0 7F 01 0E	CLR	ALLFLG	CLEAR FLAG
01A3 20 21	BRA	DIR6	
01A5 7E 02 32	JMP	ERR	REPORT ERROR
01A8 FE 70 94	LDX	BUFPNT	RESET POINTER
01AB A6 00	LDA A	0,X	GET NEXT CHAR
01AD 81 2E	CMP A	#'.'	IS IT PERIOD?
01AF 27 10	BEQ	DIR5	
01B1 BD 71 1B	JSR	CLASS	CLASSIFY IT
01B4 25 74	BCS	SERR	TERM?
01B6 81 39	CMP A	#'9	IS IT NUMBER?
01B8 22 07	BHI	DIR5	
01BA BD 71 21	JSR	NXTCH	GET NEXT
01BD 24 FB	BCC	DIR47	TERM?
01BF 20 DA	BRA	DIR42	
01C1 BD 03 47	JSR	GETNAM	INPUT NAME
01C4 25 64	BCS	SERR	ERROR?
01C6 BD 02 4B	JSR	PRDIR	GO PRINT DIR
01C9 BD 6D	BSR	TSTTRM	TEST TERM
01CB 26 F4	BNE	DIR5	
01CD CE 77 40	LDX	#FCB	POINT TO FCB
01D0 86 10	LDA A	#16	OPEN SYSTEM REC
01D2 A7 00	STA A	0,X	
01D4 BD 78 06	JSR	FMS	
01D7 26 59	BNE	ERR	ERRORS?
01D9 BD 03 2F	JSR	GETIR	GET FIRST RECORD
01DC 26 54	BNE	ERR	ERROR?
01DE CE 03 BF	LDX	#FST	POINT TO STRING
01E1 BD 71 18	JSR	PSTRNG	PRINT IT
01E4 5F	CLR B		
01E5 CE 01 14	LDX	#FMAX	
01E8 BD 71 33	JSR	OUTDEC	PRINT FILES
01EB 8D 56	BSR	OUTCOM	PRINT COMMA
01ED CE 03 C6	LDX	#SST	POINT TO STRING
01F0 C6 08	LDA B	#8	
01F2 BD 03 7B	JSR	PDATA	PRINT IT
01F5 CE 01 16	LDX	#SECS	
01F8 5F	CLR B		
01F9 BD 71 33	JSR	OUTDEC	PRINT SECTORS
01FC 8D 45	BSR	OUTCOM	PRINT COMMA
01FE CE 03 CE	LDX	#LST	POINT TO STRING
0201 C6 08	LDA B	#8	
0203 BD 03 7B	JSR	PDATA	PRINT IT
0206 CE 01 18	LDX	#LARG	
0209 5F	CLR B		
020A BD 71 33	JSR	OUTDEC	PRINT LARGEST
020D 8D 34	BSR	OUTCOM	OUTPUT COMMA
020F CE 03 D6	LDX	#FRST	POINT TO STRING
0212 C6 05	LDA B	#5	
0214 BD 03 7B	JSR	PDATA	PRINT IT
0217 5F	CLR B		
0218 CE 77 55	LDX	#FCB+21	
021B BD 71 33	JSR	OUTDEC	PRINT FREE
021E BD 71 1E	JSR	PCRLF	PRINT CR & LF
0221 7E 01 2D	JMP	DIR22	REPERT

* EXIT DIR UTILITY

0224 BD 71 1E	EXIT	JSR	PCRLF	PRINT CR & LF
0227 7E 71 03		JMP	WARMS	RETURN TO FLEX

* REPORT SYNTAX ERROR

022A CE 03 F6	SERR	LDX	#SYST	POINT TO STRING
022D BD 71 18		JSR	PSTRNG	PRINT IT
0230 20 03		BRA	ERR2	

* REPORT DISK ERROR

0232 BD 71 3C	ERR	JSR	RPTERR	REPORT ERROR
0235 7E 71 03	ERR2	JMP	WARMS	RETURN TO FLEX

* TEST FOR TERMINATOR

0238 B6 70 91	TSTTRM	LDR A	LSTTRM	CHECK LAST TERM
023B 81 0D		CMP A	#\$D	IS IT CR?
023D 27 03		BEQ	TSTTR2	
023F B1 70 82		CMP A	EOL	IS IT EOL?
0242 39	TSTTR2	RTS		

* OUTPUT COMMA AND SPACE

0243 86 2C	OUTCOM	LDA A	#',	SETUP COMMA
0245 BD 71 12		JSR	PUTCHR	PRINT IT
0248 7E 03 2A		JMP	OUTSP	DO SPACE

* PRINT DIRECTORY LIST

024B CE 00 00	PRDIR	LDX	#0	CLEAR X
024E FF 01 12		STX	FNUM	CLEAR COUNTERS
0251 FF 01 14		STX	FMAX	
0254 FF 01 16		STX	SECS	
0257 FF 01 18		STX	LARG	
025A CE 77 40		LDX	#FCB	POINT TO FCB
025D 86 06		LDR A	#6	OPEN DIRECTORY
025F A7 00		STA A	0,X	
0261 BD 78 06		JSR	FMS	
0264 26 CC		BNE	ERR	ERRORS?
0266 7C 01 0F		INC	LOOP	SET LOOP FLAG
0269 BD 71 1E	PRDIR2	JSR	PCRLF	PRINT CR & LF
026C BD 03 2F	PRDIR3	JSR	GETIR	GET INFO RECORD
026F 27 03		BEQ	PRDI32	ERRORS?
0271 7E 03 20		JMP	PRDIR6	
0274 7D 77 44	PRDI32	TST	FCB+4	NAME PRESENT?
0277 26 03		BNE	PRDI34	
0279 7E 03 29		JMP	PRDIR7	
027C FE 01 12	PRDI34	LDX	FNUM	INC FILE NUMBER
027F 08		INX		
0280 FF 01 12		STX	FNUM	

DIR UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 5

0283 7D 77 44		TST	FCB+4	DELETED FILE?
0286 2B E4		BMI	PRDIR3	
0288 B6 77 55		LDA A	FCB+21	GET FILE SIZE
028B F6 77 56		LDA B	FCB+22	
028E B1 01 18		CMP A	LARG	LARGEST?
0291 25 00		BLO	PRD345	
0293 22 05		BHI	PRD342	
0295 F1 01 19		CMP B	LARG+1	
0298 23 06		BLS	PRD345	
029A B7 01 18	PRD342	STA A	LARG	SET NEW LARGEST
029D F7 01 19		STA B	LARG+1	
02A0 FB 01 17	PRD345	ADD B	SECS+1	ADD IN SECTOR COUNT
02A3 B9 01 16		ADC A	SECS	
02A6 B7 01 16		STA A	SECS	SAVE NEW COUNT
02A9 F7 01 17		STA B	SECS+1	
02AC FE 01 14		LDX	FMAX	BUMP FILE NUMBER
02AF 08		INX		
02B0 FF 01 14		STX ^	FMAX	
02B3 CE 77 44		LDX	#FCB+4	POINT TO NAME
02B6 7D 01 0E		TST	ALLFLG	ALL FILES?
02B9 26 27		BNE	PRDIR4	
02BB FF 01 20		STX	DATPNT	SAVE POINTER
02BE CE 01 03		LDX	#NAME	TRY TO MATCH NAME
02C1 FF 01 1E		STX	INDEX	
02C4 FE 01 1E	PRDI35	LDX	INDEX	
02C7 8C 01 0E		CPX	#EXT+3	
02CA 27 16		BEQ	PRDIR4	
02CC A6 00		LDA A	0,X	
02CE 08		INX		
02CF FF 01 1E		STX	INDEX	
02D2 FE 01 20		LDX	DATPNT	
02D5 4D		TST A		CHAR NULL?
02D6 27 04		BEQ	PRDI37	
02D8 A1 00		CMP A	0,X	DO COMPARISON
02DA 26 90		BNE	PRDIR3	
02DC 08	PRDI37	INX		MOVE TO NEXT
02DD FF 01 20		STX	DATPNT	
02E0 20 E2		BRA	PRDI35	
02E2 C6 01	PRDIR4	LDA B	#1	SET FLAG
02E4 CE 01 12		LDX	#FNUM	
02E7 BD 71 33		JSR	OUTDEC	PRINT FILE NUMBER
02EA 8D 3E		BSR	OUTSP	PRINT SPACES
02EC 8D 3C		BSR	OUTSP	
02EE CE 77 44		LDX	#FCB+4	POINT TO NAME
02F1 C6 08		LDA B	#8	
02F3 BD 03 39		JSR	PRNAM	PRINT NAME
02F6 86 2E		LDA A	#1	
02F8 BD 71 12		JSR	PUTCHR	OUTPUT .
02FB C6 03		LDA B	#3	
02FD 8D 3A		BSR	PRIAM	PRINT EXT
02FF 8D 29		BSR	OUTSP	OUTPUT SPACES
0301 8D 27		BSR	OUTSP	
0303 CE 77 51		LDX	#FCB+17	OUTPUT START
0306 BD 03 89		JSR	OUTPNT	

0309 8D 1F		BSR	OUTSP	
030B 8D 1D		BSR	OUTSP	
030D CE 77 53		LDX	#FCB+19	OUTPUT END
0310 BD 03 89		JSR	OUTPNT	
0313 8D 15		BSR	OUTSP	
0315 CE 77 55		LDX	#FCB+21	
0318 C6 01		LDA B	#1	
031A BD 71 33		JSR	OUTDEC	OUTPUT SIZE
031D 7E 02 69		JMP	PRDIR2	REPEAT
0320 A6 01	PRDIR6	LDA A	1,X	GET ERROR TYPE
0322 81 08		CMP A	#8	IS IT EOF?
0324 27 03	PRDI65	BEQ	PRDIR7	
0326 7E 02 32		JMP	ERR	REPORT ERROR
0329 39	PRDIR7	RTS		

* OUTPUT SPACE

032A 86 20	OUTSP	LDA A	#SPC	SETUP SPACE
032C 7E 71 12		JMP	PUTCHR	PRINT IT

* GET INFO RECORD

032F CE 77 40	GETIR	LDX	#FCB	POINT TO FCB
0332 86 07		LDA A	#7	GET IR CODE
0334 A7 00		STA A	0,X	
0336 7E 78 06		JMP	FMS	CALL FMS

* PRINT FILE NAME

0339 A6 00	PRNAM	LDA A	0,X	GET CHARACTER
033B 26 02		BNE	PRNAM3	NULL?
033D 86 20		LDA A	#SPC	SETUP SPACE
033F BD 71 12	PRNAM3	JSR	PUTCHR	PRINT CHAR
0342 08		INX		BUMP TO NEXT
0343 5A		DEC B		
0344 26 F3		BNE	PRNAM	
0346 39		RTS		

* GET FILE MATCH NAME

0347 7F 01 0E	GETNAM	CLR	ALLFLG	CLEAR ALL
0348 CE 01 03		LDX	#NAME	POINT TO BUFFER
034D C6 0B		LDA B	#11	SET COUNT
034F 6F 00	GETNAM2	CLR	0,X	CLEAR BUFFER
0351 08		INX		
0352 5A		DEC B		
0353 26 FA		BNE	GETNAM2	
0355 FE 70 94		LDX	BUFPNT	POINT TO NAME
0358 A6 00		LDA A	0,X	GET CHARACTER
035A 81 2E		CMP A	# .	IS IT ' .' ?
035C 27 06		BEQ	GLTNAM4	
035E CE 00 FF		LDX	#NAME-4	RESET POINTER
0361 7E 71 27		JMP	GLTFIL	GET FILE NAME
0364 7C 70 95	GETNAM4	INC	BUFPNT+1	FIX POINTER

0367 C6 03		LDA B	#3	SET COUNT
0369 CE 01 0B		LDX	#EXT	POINT TO EXT
036C BD 71 21	GETNRS	JSR	NXTCH	GET CHARACTER
036F 25 08		BCS	GETNA6	TERM?
0371 A7 00		STA R	0,X	SAVE CHAR
0373 08		INX		
0374 5A		DEC B		
0375 2A F5		BPL	GETNRS	
0377 0D		SEC		
0378 39		RTS		RETURN
0379 0C	GETNA6	CLC		
037A 39		RTS		

* PRINT STRING WITHOUT CR & LF

037B A6 00	PDATA	LDA A	0,X	GET CHARACTER
037D 27 08		BEQ	PDATA2	NULL?
037F BD 71 12		JSR	PUTCHR	PRINT IT
0382 08		INX		BUMP TO NEXT
0383 5A		DEC B		DEC COUNT
0384 26 F5		BNE	PDATA	
0386 39		RTS		RETURN
0387 5F	PDATA2	CLR B		
0388 39		RTS		

* PRINT DISK ADDRESS

0389 BD 71 39	OUTPNT	JSR	OUTHEX	OUTPUT DIGIT
038C 8D 04		BSR	OUTM	PRINT '-'
038E 08		INX		
038F 7E 71 39		JMP	OUTHEX	PRINT REST

* PRINT MINUS SIGN

0392 86 2D	OUTM	LDA A	#'-'	SETUP '-'
0394 7E 71 12		JMP	PUTCHR	PRINT IT

* STRINGS

	HDR	FCC	'FILE#'	NAME	TYPE	BEGIN	END	SIZE'
0397 46		FCB	4					
03BE 04		FCC	'FILES='					
03BF 46	FST	FCC	4					
03C5 04		FCB	'SECTORS='					
03C6 53	SST	FCC	'LARGEST='					
03CE 4C	LST	FCC	'FREE='					
03D6 46	FRST	FCC	'DIRECTORY OF DRIVE NUMBER '					
03DB 44	DIRST	FCC	4					
03F5 04		FCB	'SYNTAX ERROR'					
03F6 53	SYST	FCC	4					
0402 04		FCB						
		END	DIR					

NO ERROR(S) DETECTED

INSTALL

The INSTALL utility is used as a convenient way of renaming a file with a .BIN extension to the same name but with a .CMD extension. In effect, you are 'installing' the file into the utility command set.

DESCRIPTION

The general syntax of the INSTALL command is:

```
INSTALL,<file spec>
```

where the file spec defaults to a BIN extension and to the working drive. This utility has the same affect as using 'RENAME,FILE.BIN,FILE.CMD'. An example will demonstrate its use.

```
+--+INSTALL,LOAD
```

This command line would cause the file named LOAD.BIN on the working drive to be renamed LOAD.CMD. This command is simply a time saver for those who do not like to type names twice!

```

*
* INSTALL FILE UTILITY
*
* COPYRIGHT (C) 1978 BY
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*

```

```
* EQUATES
```

7806	FMS	EQU	\$7806	X3406
7803	FMSCLS	EQU	\$7803	X3403
7103	WARMs	EQU	\$7103	XAD03
710F	GETCHR	EQU	\$710F	XAD15
7112	PUTCHR	EQU	\$7112	XAD16
7118	PSTRNG	EQU	\$7118	XAD1E
711E	PCRLF	EQU	\$711E	XAD24
7127	GETFIL	EQU	\$7127	XAD2D
712D	SETEXT	EQU	\$712D	XAD33
713C	RPTERR	EQU	\$713C	XAD3F
7740	FCB	EQU	\$7740	XAC40
7600		ORG	\$7600	X410C
7600 20 01	INST	BRA	INST1	
7602 01	VN	FCB	1	VERSION NUMBER
7603 CE 77 40	INST1	LDX	#FCB	POINT TO FCB
7606 BD 71 27		JSR	GETFIL	GET FILE NAME
7609 25 2D		BCS	INST4	ERROR?
760B 4F		CLR A		SET BIN CODE
760C CE 77 40		LDX	#FCB	SET EXTENSION
760F BD 71 2D		JSR	SETEXT	POINT TO NAME
7612 CE 77 44		LDX	#FCB+4	SET COUNT
7615 C6 08		LDA B	#8	SET CHARACTER
7617 A6 00	INST3	LDA R	0,X	MOVE TO WB
7619 A7 31		STA R	49,X	BUMP TO NEXT
761B 08		INX		DEC THE COUNT
761C 5A		DEC B		REPEAT
761D 26 F8		BNE	INST3	CLEAR EXTENSION
761F 6F 31		CLR	49,X	POINT TO 2ND NAME
7621 CE 77 71		LDX	#FCB+49	CMD EXTENSION
7624 86 02		LDA R	#2	SET EXTENSION
7626 BD 71 2D		JSR	SETEXT	POINT TO FCB
7629 CE 77 40		LDX	#FCB	SET FOR RENAME
762C 86 0D		LDA R	#13	CALL FMS
762E A7 00		STA R	0,X	ERRORS?
7630 BD 78 06		JSR	FMS	RETURN TO FLEX
7633 26 0C		BNE	INST6	
7635 7E 71 03		JMP	WARMs	

INSTALL FILE UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 2

7638 CE 76 46	INST4	LDX	#ILST	POINT TO STRING
763B BD 71 18		JSR	PSTRNG	PRINT IT
763E 7E 71 03	INST5	JMP	WARMS	RETURN
7641 BD 71 3C	INST6	JSR	RPTERR	
7644 20 F8		BRA	INST5	

* STRINGS

7646 49		ILST	FCC	'ILLEGAL FILE NAME'
7657 04			FCB	4
		END		INST

NO ERROR(S) DETECTED

SYMBOL TABLE:

FCB	7740	FMS	7806	FMSCLS	7803	GETCHR	710F	GETFIL	7127
ILST	7646	INST	7600	INST1	7603	INST3	7617	INST4	7638
INST5	763E	INST6	7641	PCRLF	711E	PSTRNG	7110	PUTCHR	7112
RPTERR	713C	SETEXT	7120	VN	7602	WARMS	7103		

FREE

The FREE command is used to report the total number of free (available) sectors on a diskette. The approximate number of kilobytes remaining is also reported.

DESCRIPTION

The general syntax of the FREE command is:

```
FREE[,<drive number>]
```

If the drive number is not specified it will default to the working drive. An example follows:

```
+++FREE,1
```

This command line will report the number of available sectors and approximate number of kilobytes remaining on the disk in drive 1.

```

*
* FREE SPACE UTILITY
*
* COPYRIGHT (C) 1978 BY
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*

```

```
* EQUATES
```

7806	FMS	EQU	\$7806	þB406
7803	FMSCLS	EQU	\$7803	þB403
7103	WARMS	EQU	\$7103	þA003
710F	GETCHR	EQU	\$710F	þAD15
7112	PUTCHR	EQU	\$7112	þAD10
7118	PSTRNG	EQU	\$7118	þAD1E
711E	PCRLF	EQU	\$711E	þAD24
7127	GETFIL	EQU	\$7127	þAD2D
712D	SETEXT	EQU	\$712D	þA033
713C	RPTERR	EQU	\$713C	þAD3F
7740	FCB	EQU	\$7740	þA0340
713F	GETHEX	EQU	\$713F	þA042
7133	OUTDEC	EQU	\$7133	þA034
708C	WASN	EQU	\$708C	þAC0C

```
* PROGRAM STARTS HERE
```

7600		ORG	\$7600	þA100
7600 20 03	FREE	BRA	FREE1	
7602 01	VN	FCB	1	VERSION NUMBER
7603 00 00	TEMP	FDB	0	
7605 BD 71 3F	FREE1	JSR	GETHEX	GET DRIVE
7608 25 51		BCS	FREE8	ERROR?
760A 5D		TST B		ANY NUMBER FOUND?
760B 26 05		BNE	FREE2	
760D B6 70 8C		LDA A	WASN	GET ASSIGNED DRIVE
7610 20 06		BRA	FREE4	
7612 FF 76 03	FREE2	STX	TEMP	SAVE NUMBER
7615 B6 76 04		LDA A	TEMP+1	GET LSB
7618 B7 77 43	FREE4	STA A	FCB+3	SET DRIVE NUMBER
761B CE 77 40		LDX	#FCB	POINT TO FCB
761E 86 10		LDA A	#16	OPEN SIR
7620 A7 00		STA A	0,X	
7622 BD 78 06		JSR	FMS	CALL FMS
7625 26 2E		BNE	FREE6	ERRORS?
7627 86 07		LDA A	#7	GET IR CODE
7629 A7 00		STA A	0,X	SET CODE
762B BD 78 06		JSR	FMS	CALL FMS
762E 26 25		BNE	FREE6	ERRORS?

7630	CE	76	64		LDX	#ST1	POINT TO STRING
7633	BD	71	18		JSR	PSTRNG	PRINT IT
7636	5F			CLR B			
7637	CE	77	55		LDX	#FCB+21	POINT TO NUMBER
763A	BD	71	33		JSR	OUTDEC	OUTPUT NUMBER
763D	CE	76	79		LDX	#ST2	POINT TO STRING
7640	BD	71	18		JSR	PSTRNG	PRINT IT
7643	CE	77	55		LDX	#FCB+21	POINT TO NUMBER
7646	C6	03			LDA B	#3	FIX COUNT
7648	64	00		LOOP	LSR	0,X	
764A	66	01			ROR	1,X	
764C	5A				DEC B		DEC THE COUNT
764D	26	F9			BNE	LOOP	
764F	BD	71	33		JSR	OUTDEC	OUTPUT NUMBER
7652	7E	71	03		JMP	WARMS	RETURN TO DOS
7655	BD	71	3C	FREE6	JSR	RPTERR	REPORT ERROR
7658	7E	71	03		JMP	WARMS	
765B	CE	76	92	FREE8	LDX	#ILD	POINT TO STRING
765E	BD	71	18		JSR	PSTRNG	PRINT IT
7661	7E	71	03		JMP	WARMS	

* STRINGS

7664 53	ST1	FCC	'SECTORS REMAINING = '
7678 04		FCB	4
7679 41	ST2	FCC	'APPROXIMATE KILOBYTES = '
7691 04		FCB	4
7692 49	ILD	FCC	'ILLEGAL DRIVE NUMBER'
76A6 04		FCB	4
	END	FREE	'

NO ERROR(S) DETECTED

SYMBOL TABLE:

REPLACE

The REPLACE command will effectively replace one file on a disk by another, deleting the first file. This command is simply a time saver.

DESCRIPTION

The general syntax of the REPLACE command is:

```
REPLACE,<file spec 1>,<file spec 2>
```

where file spec 1 is the name of the file to be deleted and file spec 2 will be renamed to that of file spec 1. The file specs default to a TXT extension and to the working drive. This command has the same affect as the two command sequence:

```
+--+DELETE,FILE1  
+--+RENAME,FILE2,FILE1
```

which effectively replaces FILE1 with FILE2. This can be performed with REPLACE as follows:

```
+--+REPLACE,FILE1,FILE2
```

which will cause FILE1.TXT to be deleted and FILE2.TXT to be renamed FILE1.TXT.

```

*
* REPLACE FILE UTILITY
*
*
* COPYRIGHT (C) 1978 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*

```

* GLOBAL VARIABLES

7740	FCB	EQU	\$7740	A040
7803	FMSCLS	EQU	\$7803	B403
7806	FMS	EQU	\$7806	B406
7103	WARMS	EQU	\$7103	A003
710F	GETCHR	EQU	\$710F	A045
7112	PUTCHR	EQU	\$7112	A010
7118	PSTRNG	EQU	\$7118	A01E
711E	PCRLF	EQU	\$711E	A024
7121	NXTCH	EQU	\$7121	A027
7127	GETFIL	EQU	\$7127	A02D
712D	SETEXT	EQU	\$712D	A033
7133	OUTDEC	EQU	\$7133	A034
713C	RPTERR	EQU	\$713C	A03F
713F	GETHEX	EQU	\$713F	A042

* PROGRAM HERE

0100		ORG	\$0100	A100
0100 20 07	REPLAC	BRA	REPLA2	

* TEMPS

0102 01	VN	FCB	1	
0103 00	QCNT2	FCB	0	
0104 00	DRV	FCB	0	
0105 00 00	TEMP1	FDB	0	
0107 00 00	TEMP2	FDB	0	
0109 CE 77 40	REPLA2	LDX	#FCB	POINT TO FCB
010C BD 71 27		JSR	GETFIL	GET FILE NAME
010F 25 0D		BCS	REPL23	ERROR?
0111 86 01		LDA A	#1	SET EXTENSION
0113 BD 71 2D		JSR	SETEXT	
0116 CE 02 6F		LDX	#FCB2	POINT TO 2ND
0119 BD 71 27		JSR	GETFIL	GET FILE NAME
011C 24 03		BCC	REPL25	ERROR?
011E 7E 01 E1	REPL23	JMP	ERR0R6	REPORT ERROR

FILE REPLACE UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 2

0121 6D 0C	REPL25	TST	12,X	TEST FOR EXT
0123 26 0C		BNE	REPLA3	
0125 FE 77 4C		LDX	FCB+12	
0128 B6 77 4E		LDA A	FCB+14	GET EXT
012B FF 02 7B		STX	FCB2+12	PUT EXT
012E B7 02 7D		STA A	FCB2+14	
0131 86 01	REPLA3	LDA A	#1	OPEN FOR READ
0133 CE 77 40		LDX	#FCB	POINT TO FCB
0136 A7 00		STA A	0,X	
0138 BD 78 06		JSR	FMS	CALL FMS
013B 26 1A		BNE	REPL33	ERROR?
013D 86 04		LDA A	#4	CLOSE FILE
013F A7 00		STA A	0,X	
0141 BD 78 06		JSR	FMS	CALL FMS
0144 26 11		BNE	REPL33	ERROR?
0146 CE 02 6F		LDX	#FCB2	POINT TO 2ND FCB
0149 B6 77 43		LDA A	FCB+3	GET DRIVE NO
014C A7 03		STA A	3,X	
014E 86 01		LDA A	#1	OPEN FOR READ
0150 A7 00		STA A	0,X	
0152 BD 78 06		JSR	FMS	CALL FMS
0155 27 03		BEQ	REPL35	ERROR?
0157 7E 01 C8	REPL33	JMP	REPLA5	REPORT ERROR
015A 86 04	REPL35	LDA A	#4	CLOSE FILE
015C A7 00		STA A	0,X	
015E BD 78 06		JSR	FMS	CALL FMS
0161 26 F4		BNE	REPL33	ERRORS?
0163 CE 02 1B		LDX	#DELST	POINT TO STRING
0166 BD 71 1E		JSR	PCRLF	PRINT CR & LF
0169 BD 71 18		JSR	PSTRNG	PRINT STRING
016C CE 77 42		LDX	#FCB+2	POINT TO FILE SPEC
016F BD 01 ED		JSR	PNAME	OUTPUT IT
0172 CE 02 22		LDX	#QUQUST	POINT TO STRING
0175 C6 12		LDA B	#18	SET COUNT
0177 BD 02 0F		JSR	PNAME4	PRINT STRING
017A CE 02 35		LDX	#SURES	POINT TO STRING
017D BD 71 18		JSR	PSTRNG	PRINT IT
0180 BD 71 0F		JSR	GETCHR	GET Y-N RESPONSE
0183 84 5F		AND A	#\$5F	MAKE UPPER
0185 81 59		CMP A	#^Y	IS IT Y?
0187 26 61		BNE	REPLA6	
0189 CE 77 40	REPLA4	LDX	#FCB	POINT TO FCB
018C 86 0C		LDA A	#12	SET FOR DELETE
018E A7 00		STA A	0,X	
0190 BD 78 06		JSR	FMS	CALL FMS
0193 26 33		BNE	REPLA5	ERROR?
0195 A6 24		LDA A	36,X	SET 1ST CHAR
0197 A7 04		STA A	4,X	RESTORE IT
0199 CE 77 44		LDX	#FCB+4	POINT TO NAME
019C FF 01 05		STX	TEMP1	SET UP POINTERS
019F CE 02 A4		LDX	#FCB2+53	
01A2 FF 01 07		STX	TEMP2	
01A5 C6 0B		LDA B	#11	SET COUNTER
01A7 FE 01 05	MOVE	LDX	TEMP1	GET POINTER

01RR A6 00		LDA A	0,X	GET CHARACTER
01AC 08		INX		BUMP TO NEXT
01AD FF 01 05		STX	TEMP1	SAVE IT
01B0 FE 01 07		LDX	TEMP2	GET DEST POINTER
01B3 A7 00		STA A	0,X	PUT CHARACTER
01B5 08		INX		BUMP TO NEXT
01B6 FF 01 07		STX	TEMP2	SAVE POINTER
01B9 5A		DEC B		DEC THE COUNTER
01BA 26 EB		BNE	MOVE	FINISHED?
01BC CE 02 6F		LDX	#FCB2	POINT TO FCB
01BF 86 0D		LDA A	#13	SET FOR RENAME
01C1 A7 00		STA A	0,X	
01C3 BD 78 06		JSR	FMS	CALL FMS
01C6 27 22		BEQ	REPLAS	ERRORS?
01C8 A6 01	REPLAS	LDA A	1,X	GET ERROR TYPE
01CA 81 03		CMP A	#3	FILE EXISTS?
01CC 27 09		BEQ	ERROR2	
01CE 81 04		CMP A	#4	NO FILE?
01D0 27 0A		BEQ	ERROR4	
01D2 BD 71 3C		JSR	RPTERR	REPORT ERROR
01D5 20 10		BRA	REPL55	
01D7 CE 02 63	ERROR2	LDX	#FEST	POINT TO STRING
01DA 20 08		BRA	ERROR7	
01DC CE 02 44	ERROR4	LDX	#NSST	POINT TO STRING
01DF 20 03		BRA	ERROR7	
01E1 CE 02 51	ERROR6	LDX	#ILST	POINT TO STRING
01E4 BD 71 18	ERROR7	JSR	PSTRNG	PRINT STRING
01E7 BD 78 03	REPL55	JSR	FMSCLS	CLOSE ALL FILES
01ER 7E 71 03	REPLAS	JMP	WARM	RETURN TO FLEX

* PRINT FILE SPEC ROUTINE

01ED 6F 00		PNAME	CLR	0,X	CLEAR MSB
01EF 5F			CLR B		CLEAR FLAG
01F0 BD 71 33			JSR	OUTDEC	PRINT DRIVE NUM
01F3 CE 77 44			LDX	#FCB+4	POINT TO NAME
01F6 86 2E			LDA A	#1	OUTPUT '.'
01F8 BD 71 12			JSR	PUTCHR	
01FB C6 08			LDA B	#8	SET COUNTER
01FD A6 00	PNAME2		LDA A	0,X	GET CHARACTER
01FF 27 03			BEQ	PNAME3	IS IT NULL?
0201 BD 71 12			JSR	PUTCHR	OUTPUT IT
0204 08	PNAME3		INX		BUMP TO NEXT
0205 5A			DEC B		DEC THE COINTER
0206 26 F5			BNE	PNAME2	REPEAT?
0208 C6 03			LDR B	#3	SET COUNT
020A 86 2E			LDA A	#1	OUTPUT '.'
020C BD 71 12			JSR	PUTCHR	
020F A6 00	PNAME4		LDA A	0,X	GET CHARACTER
0211 27 07			BEQ	PNAME6	IS IT NULL?
0213 BD 71 12			JSR	PUTCHR	OUTPUT IT
0216 08			INX		BUMP TO NEXT
0217 5A			DEC B		DEC THE COUNTER
0218 26 F5			BNE	PNAME4	

021A 39	PNAME6 RTS	RETURN
* STRINGS		
021B 46	DELST FCC	'FILE "
0221 04	FCB	4
0222 22	QUQUST FCC	'" WILL BE DELETED!"
0234 04	FCB	4
0235 41	SURES FCC	'ARE YOU SURE? '
0243 04	FCB	4
0244 4E	NSST FCC	'NO SUCH FILE'
0250 04	FCB	4
0251 49	ILST FCC	'ILLEGAL FILE NAME'
0262 04	FCB	4
0263 46	FEST FCC	'FILE EXISTS'
026E 04	FCB	4
026F FCB2	RMB 192	3 20
	END	REPLAC

NO ERROR(S) DETECTED

SYMBOL TABLE:

DELST 021B	DRV 0104	ERROR2 01D7	ERROR4 01DC	ERROR6 01E1
ERROR7 01E4	FCB 7740	FCB2 026F	FEST 0263	FMS 7806
FMSCLS 7803	GETCHR 710F	GETFIL 7127	GETHEX 713F	ILST 0251
MOVE 01A7	NSST 0244	NXTCH 7121	OUTDEC 7133	PCRLF 711E
PNAME 01ED	PNAME2 01FD	PNAME3 0204	PNAME4 020F	PNAME6 021A
PSTRNG 7118	PUTCHR 7112	QCNT2 0103	QUQUST 0222	REPL23 011E
REPL25 0121	REPL33 0157	REPL35 015A	REPL55 01E7	REPLA2 0109
REPLA3 0131	REPLA4 0189	REPLAS 01C8	REPLA6 01EA	REPLAC 0100
RPTERR 713C	SETEXT 712D	SURES 0235	TEMP1 0105	TEMP2 0107
VN 0102	WARMS 7103			

TEST

The TEST command is used for testing all sectors on a diskette. Any bad sectors found will be reported to the terminal.

DESCRIPTION

The general syntax of the TEST command is:

```
TEST[,<drive number>]
```

where the drive number specifies which disk is to be tested and defaults to the working drive. Any sectors found to be bad during the test are reported to the terminal in the form of two hex numbers, the first representing the track number, the second is the sector. An example follows:

```
+++TEST,0
```

This will test the diskette in drive 0 for bad sectors. It should be noted that TEST requires a moderat amount of time to run since all data on the disk is read.

```

*
* TEST DISK FOR ERRORS
*
* COPYRIGHT 1978 BY
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*

```

7600

ORG \$7600

A100

* SYSTEM EQUATES

7806	FMS	EQU	\$7806	A4C6
7740	FCB	EQU	\$7740	A4C4
713F	GETHEX	EQU	\$713F	A4C2
711E	PCRLF	EQU	\$711E	A4C4
AD45	OUTADR	EQU	\$AD45	A4C7
7118	PSTRNG	EQU	\$7118	A4C6
7103	WARMS	EQU	\$7103	A4C3
708C	WASN	EQU	\$708C	A4C0

* PROGRAM STARTS HERE

7600 20 05 TEST BRA TEST1

VERSION NUMBER

7602 01	VN	FCB	1
7603 00	TRACK	FCB	0
7604 02	SECTOR	FCB	2
7605 00 00	TEMP	FDB	0

7607 BD 71 3F	TEST1	JSR GETHEX	GETDRIVE NUMBER
760A 25 37		BGS TEST9	ERROR?
760C FF 76 05		STX TEMP	SAVE VALUE
760F CE 77 40		LDX #FCB	POINT TO FCB
7612 6F 02		CLR 2,X	CLEAR STATUS
7614 B6 76 06		LDA A TEMP+1	GET DRIVE NUMBER
7617 5D		TST B	CHECK FOR DRIVE
7618 26 03		BNE TEST15	
761A B6 70 8C		LDA A WASN	GET WORK DRIVE
761D A7 03	TEST15	STA A 3,X	SET IN FCB
761F 86 09		LDA A #9	SET FOR READ
7621 A7 00		STA A 0,X	
7623 8D 26	TEST2	BSR SET	SET TRACK - SECTOR
7625 25 13		BGS TEST8	FINISHED?
7627 BD 78 06		JSR FMS	READ SECTOR
762A 27 F7		BEQ TEST2	ERRORS?
762C CE 76 77		LDX #STR	POINT TO STRING
762F BD 71 18		JSR PSTRNG	PRINT IT
7632 CE 76 03		LDX #TRACK	POINT TO TRACK
7635 BD AD 45		JSR OUTADR	PRINT IT
7638 20 E9		BRA TEST2	
763A CE 76 9B	TEST8	LDX #FINST	POINT TO STRING

TEST DISK UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 2

763D BD 71 18	JSR	PSTRNG	PRINT IT	
7640 ?E 71 03	JMP	WARMS	RETURN TO FLEX	
7643 CE 76 86	TEST85	LDX	#ILST	POINT TO STRING
7646 BD 71 18	JSR	PSTRNG	PRINT IT	
7649 20 F5	BRA	TEST85		

* SET TRACK AND SECTOR

764B	CE	77	40	SET	LDX	#FCB	POINT TO FCB
764E	B6	76	04		LDA A	SECTOR	GET SECTOR
7651	81	12			CMP A	#16	
7653	27	05			BEQ	SET2	
7655	7C	76	04		INC	SECTOR	BUMP TO NEXT
7658	20	0F			BRA	SET4	
765A	B6	01		SET2	LDA A	#1	RESTORE TO 1
765C	B7	76	04		STA A	SECTOR	
765F	B6	76	03		LDA A	TRACK	GET TRACK
7662	81	22			CMP A	#34	MAX TRACK?
7664	27	0F			BEQ	SET6	
7666	7C	76	03		INC	TRACK	BUMP TRACK
7669	B6	76	04	SET4	LDA A	SECTOR	
766C	A7	1F			STA A	31,X	SET IN FCB
766E	B6	76	03		LDA A	TRACK	
7671	A7	1E			STA A	30,X	
7673	0C				CLC		SHOW NO END
7674	39				RTS		RETURN
7675	0D			SET6	SEC		SHOW END
7676	39				RTS		
7677	42			STR	FCC	'BAD SECTOR AT '	
7685	04				FCB	4	
7686	49			ILST	FCC	'ILLEGAL DRIVE NUMBER'	
769A	04				FCB	4	
769B	54			FNST	FCC	'TEST COMPLETED'	
76A9	04				FCB	4	
				END	TEST		

NO ERRORS(S) DETECTED

SYMBOL TABLE:

FCB	7740	FMS	7806	FNST	769B	GETHEX	713F	ILST	7686
OUTADR	AD45	PCRLF	711E	PSTRNG	7118	SECTOR	7604	SET	7648
SET2	765A	SET4	7669	SET6	7675	STR	7677	TEMP	7605
TEST	7600	TEST1	7607	TEST15	7610	TEST2	7623	TEST8	763A
TEST85	7640	TEST9	7643	TRACK	7603	VN	7602	WHRMS	7103
WASN	788C								

FILES

The FILES utility is similar to the CAT command but displays only the file names and extensions. This command is useful for getting a short and quick report of the directory contents.

DESCRIPTION

The general syntax of the FILES command is:

```
FILES[,<drive list>][,<match list>]
```

where <drive list> and <match list> are the same as described in the CAT command. The file names will be listed across the page and in a columnar fashion. The number displayed per line is determined by the TTYSET Width parameter. If the Width is zero, 80 columns are assumed to be available and 5 names will be listed on each line. Smaller Width values will result in fewer names per line being displayed. A few examples follow:

```
+++FILES  
+++FILES,1,A.T,FR
```

The first example would list all files on the working drive. The second example would list only those files on drive 1 whose names began with 'A' and extensions began with 'T', as well as those files whose names started with 'FR'.

```

* FILES UTILITY PROGRAM
*
* COPYRIGHT (C) 1978 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*
* GLOBAL VARIABLES

```

7082	EOL	EQU	\$7082	†AC02
7084	WIDTH	EQU	\$7084	†AC04
7740	FCB	EQU	\$7740	†A240
708C	WASN	EQU	\$708C	†AC1C
708C	ASN	EQU	WASN	
7091	LSTTRM	EQU	\$7091	†AC11
7094	BUFPNT	EQU	\$7094	†AC14
0020	SPC	EQU	\$20	?
7803	FMSCLS	EQU	\$7803	†B403
7806	FMS	EQU	\$7806	†B406
7103	WARM8	EQU	\$7103	†AN03
710F	GETCHR	EQU	\$710F	†AN15
7112	PUTCHR	EQU	\$7112	†AN18
7118	PSTRNG	EQU	\$7118	†AN1E
711B	CLASS	EQU	\$711B	†AN24
711E	PCRLF	EQU	\$711E	†AN24
7121	NXTCH	EQU	\$7121	†AN27
7127	GETFIL	EQU	\$7127	†AN2A
7133	OUTDEC	EQU	\$7133	†AN37
7139	OUTHEX	EQU	\$7139	†AN3E
713C	RPTERR	EQU	\$713C	†AN3F
713F	GETHEX	EQU	\$713F	†AN42
0100		ORG	\$0100	†A100
0100 20 17	FIL	BRA	FIL1	
0102 01	VN	FCB	1	VERSION NUMBER
0103 00	NRME	FCB	0, 0, 0, 0	
0107 00		FCB	0, 0, 0, 0	
010B 00	EXT	FCB	0, 0, 0	
010E 00	ALLFLG	FCB	0	
010F 00 00	INDEX	FDB	0	
0111 00 00	DATPNT	FDB	0	
0113 00	LOOP	FCB	0	
0114 00 00	POINT	FDB	0	
0116 00	COUNT	FCB	0	
0117 00	COL	FCB	0	
0118 00	NPL	FCB	0	

* PROGRAM HERE

0119 FE 70 94	FIL1	LDX	BUFPNT	POINT TO BUFFER
011C FF 01 14		STX	POINT	SAVE POS
011F B6 70 84		LDA A	WIDTH	GET WIDTH
0122 27 06		BEQ	FIL12	SET?
0124 47		ASR A		DIVIDE BY 16
0125 47		ASR A		
0126 47		ASR A		
0127 47		ASR A		
0128 20 02		BRA	FIL15	
012A 86 05	FIL12	LDA A	#5	SET DEFAULT
012C B7 01 18	FIL15	STA A	NPL	SET NUM PER LINE
012F BD 71 1E		JSR	PCRLF	PRINT CR & LF
0132 FE 01 14	FIL22	LDX	POINT	GET POINTER
0135 A6 00		LDA A	0, X	GET CHARACTER
0137 81 2E		CMP A	#'.'	IS IT PERIOD?
0139 27 09		BEQ	FIL25	
013B BD 71 1B		JSR	CLASS	CLASSIFY IT
013E 25 04		BCS	FIL25	TERM?
0140 81 39		CMP A	#'9	IS IT NUMBER?
0142 23 12		BLS	FIL3	
0144 7D 01 13	FIL25	TST	LOOP	LOOPING?
0147 27 03		BEQ	FIL29	
0149 7E 71 03	FIL27	JMP	WARM	RETURN TO FLEX
014C CE 77 40	FIL29	LDX	#FCB	POINT TO FCB
014F B6 70 8C		LDA A	ASN	GET DRIVE
0152 A7 03		STA A	3, X	SET IN FCB
0154 20 1E		BRA	FIL4	
0156 FE 01 14	FIL3	LDX	POINT	RESTORE POINTER
0159 FF 70 94		STX	BUFPNT	
015C BD 71 21		JSR	NXTCH	GET NEXT CHAR
015F 84 03		AND A	#3	MASK NUMBER
0161 CE 77 40		LDX	#FCB	POINT TO FCB
0164 A7 03		STA A	3, X	SAVE IN DRIVE
0166 BD 71 21		JSR	NXTCH	GET NEXT CHAR
0169 25 03		BCS	FIL35	TERM?
016B 7E 01 D5		JMP	SERR	IF NOT, ERROR
016E FE 70 94	FIL35	LDX	BUFPNT	SET POINTER
0171 FF 01 14		STX	POINT	
0174 CE 77 40	FIL4	LDX	#FCB	POINT TO FCB
0177 86 10		LDA A	#16	OPEN SYSTEM REC
0179 A7 00		STA A	0, X	
017B BD 70 06		JSR	FMS	
017E 26 21		BNE	FIL43	ERROR?
0180 BD 02 94		JSR	GETIR	GET FIRST RECORD
0183 26 10		BNE	FIL43	ERROR?
0185 CE 02 EF		LDX	#FILEST	POINT TO STRING
0188 BD 71 18		JSR	PSTRNG	PRINT IT
018B CE 77 42		LDX	#FCB+2	POINT TO DRIVE
0192 6F 00		CLR	0, X	
0190 5F		CLR B		
0191 BD 71 33		JSR	OUTDEC	PRINT IT

0194 BD 71 1E		JSR	PCRLF	PRINT CR & LF
0197 BD 01 E3	FIL42	JSR	TSTTRM	TEST TERMINATOR
019A 26 08		BNE	FIL45	
019C 7F 01 0E		CLR	ALLFLG	CLEAR FLAG
019F 20 21		BRA	FILE6	
01A1 7E 01 DD	FIL43	JMP	ERR	REPORT ERROR
01A4 FE 70 94	FIL45	LDX	BUFPNT	RESET POINTER
01A7 A6 00		LDA A	0,X	GET NEXT CHAR
01A9 81 2E		CMP A	#1.	IS IT PERIOD?
01AB 27 10		BEQ	FIL5	
01AD BD 71 1B		JSR	CLASS	CLASSIFY IT
01B0 25 23		BCS	SERR	TERM?
01B2 81 39		CMP A	#'9	IS IT NUMBER?
01B4 22 07		BHI	FIL5	
01B6 BD 71 21	FIL47	JSR	NXTCH	GET NEXT
01B9 24 FB		BCC	FIL47	TERM?
01BB 20 DA		BRA	FIL42	
01BD BD 02 AD	FIL5	JSR	GETNAM	INPUT NAME
01C0 25 13		BCS	SERR	ERROR?
01C2 BD 01 EE	FIL6	JSR	PRDIR	GO PRINT DIR
01C5 8D 1C		BSR	TSTTRM	TEST TERM
01C7 26 F4		BNE	FIL5	
01C9 BD 71 1E		JSR	PCRLF	PRINT CR & LF
01CC 7E 01 32		JMP	FIL22	REPEAT

* EXIT DIR UTILITY

01CF BD 71 1E	EXIT	JSR	PCRLF	PRINT CR & LF
01D2 7E 71 03		JMP	WARMS	RETURN TO FLEX

* REPORT SYNTAX ERROR

01D5 CE 03 06	SERR	LDX	#SYST	POINT TO STRING
01D8 BD 71 18		JSR	PSTRNG	PRINT IT
01DB 20 03		BRA	ERR2	

* REPORT DISK ERROR

01DD BD 71 3C	ERR	JSR	RPTERR	REPORT ERROR
01E0 7E 71 03	ERR2	JMP	WARMS	RETURN TO FLEX

* TEST FOR TERMINATOR

01E3 B6 70 91	TSTTRM	LDA A	LSTTRM	CHECK LAST TERM
01E6 81 0D		CMP A	#\$D	IS IT CR?
01E8 27 03		BEQ	TSTTR2	
01EA B1 70 82		CMP A	EOL	IS IT EOL?
01ED 39	TSTTR2	RTS		

* PRINT DIRECTORY LIST

01EE 7F 01 17	PRDIR	CLR	COL	
01F1 CE 77 40		LDX	#FCB	POINT TO FCB
01F4 86 06		LDA A	#6	OPEN DIRECTORY

01F6 A7 00		STA A 0,X	
01F8 BD 78 06		JSR FMS	
01FB 26 E0		BNE ERR	ERRORS?
01FD 7C 01 13		INC LOOP	SET LOOP FLAG
0200 BD 71 1E	PRDIR2	JSR PCRLF	PRINT CR & LF
0203 BD 02 94	PRDIR3	JSR GETIR	GET INFO RECORD
0206 27 03		BEQ PRDI32	ERRORS?
0208 7E 02 8A		JMP PRDIR6	
020B CE 77 40	PRDI32	LDX #FCB	POINT TO FCB
020E 6D 04		TST 4,X	NAME PRESENT?
0210 26 0B		BNE PRDI34	
0212 7D 01 17		TST COL	CHECK COLUMN
0215 27 03		BEQ PRDI33	
0217 BD 71 1E		JSR PCRLF	PRINT CR & LF
021A 7E 02 93	PRDI33	JMP PRDIR7	
021D 2B E4	PRDI34	BMI PRDIR3	DELETED NAME?
021F A6 0F		LDA A 15,X	CHECK C PROT
0221 84 10		AND A #\$10	
0223 26 DE		BNE PRDIR3	
0225 CE 77 44		LDX #FCB+4	POINT TO NAME
0228 7D 01 0E		TST ALLFLG	ALL FILES?
022B 26 27		BNE PRDIR4	
022D FF 01 11		STX DATPNT	SAVE POINTER
0230 CE 01 03		LDX #NAME	TRY TO MATCH NAME
0233 FF 01 0F		STX INDEX	
0236 FE 01 0F	PRDI35	LDX INDEX	
0239 8C 01 0E		CPX #EXT+3	
023C 27 16		BEQ PRDIR4	
023E A6 00		LDA A 0,X	
0240 08		INX	
0241 FF 01 0F		STX INDEX	
0244 FE 01 11		LDX DATPNT	
0247 4D		TST A	CHAR NULL?
0248 27 04		BEQ PRDI37	
024A A1 00		CMP A 0,X	DO COMPARISON
024C 26 B5		BNE PRDIR3	
024E 08	PRDI37	INX	MOVE TO NEXT
024F FF 01 11		STX DATPNT	
0252 20 E2		BRA PRDI35	
0254 CE 77 44	PRDIR4	LDX #FCB+4	POINT TO NAME
0257 7F 01 16		CLR COUNT	
025A C6 08		LDA B #8	SET COUNTER
025C BD 02 9E		JSR PRNAM	PRINT NAME
025F 86 2E		LDA A #1	OUTPUT PERIOD
0261 BD 71 12		JSR PUTCHR	
0264 C6 03		LDA B #3	SET COUNT
0266 BD 02 9E		JSR PRNAM	PRINT EXT
0269 7C 01 17		INC COL	BUMP COLUMN COUNT
026C B6 01 17		LDA A COL	GET COUNT
026F B1 01 18		CMP A NPL	MAX YET?
0272 26 06		BNE PRDI45	
0274 7F 01 17		CLR COL	CLEAR COUNT
0277 7E 02 00		JMP PRDIR2	REPEAT
027A C6 0F	PRDI45	LDA B #15	FILL SPACES

027C F0 01 16		SUB B	COUNT	
027F 86 20	PRDIR5	LDA A	#\$20	SETUP SPACE
0281 BD 71 12		JSR	PUTCHR	OUTPUT IT
0284 5A		DEC B		DEC THE COUNT
0285 26 F8		BNE	PRDIR5	
0287 7E 02 03		JMP	PRDIR3	
028A A6 01	PRDIR6	LDA A	1,X	GET ERROR TYPE
028C 81 08		CMP A	#0	IS IT EOF?
028E 27 03	PRDI65	BEQ	PRDIR7	
0290 7E 01 DD		JMP	ERR	REPORT ERROR
0293 39	PRDIR7	RTS		

* GET INFO RECORD

0294 CE 77 40	GETIR	LDX	#FCB	POINT TO FCB
0297 86 07		LDA A	#7	GET IR CODE
0299 A7 00		STA A	0,X	
029B 7E 78 06		JMP	FMS	CALL FMS

* PRINT FILE NAME

029E A6 00	PRNAM	LDA A	0,X	GET CHARACTER
02A0 27 06		BEQ	PRNAM2	
02A2 BD 71 12		JSR	PUTCHR	OUTPUT CHAR
02A5 7C 01 16		INC	COUNT	BUMP COUNT
02A8 08	PRNAM2	INX		BUMP TO NEXT
02A9 5A		DEC B		
02AA 26 F2		BNE	PRNAM	
02AC 39		RTS		RETURN

* GET FILE MATCH NAME

02AD 7F 01 0E	GETNAM	CLR	ALLFLG	CLEAR ALL
02B0 CE 01 03		LDX	#NAME	POINT TO BUFFER
02B3 C6 0B		LDA B	#11	SET COUNT
02B5 6F 00	GETNA2	CLR	0,X	CLEAR BUFFER
02B7 08		INX		
02B9 5A		DEC B		
02B9 26 FA		BNE	GETNA2	
02BB FE 70 94		LDX	BUFPNT	POINT TO NAME
02BE A6 00		LDA A	0,X	GET CHARACTER
02C0 81 2E		CMP A	#`.	IS IT `.'
02C2 27 06		BEQ	GETNA4	
02C4 CE 00 FF		LDX	#NAME-4	RESET POINTER
02C7 7E 71 27		JMP	GETFIL	GET FILE NAME
02CA 7C 70 95	GETNA4	INC	BUFPNT+1	FIX POINTER
02CD C6 03		LDA B	#3	SET COUNT
02CF CE 01 0B		LDX	#EXT	POINT TO EXT
02D2 BD 71 21	GETNA5	JSR	NXTCH	GET CHARACTER
02D5 25 08		BCS	GETNA6	TERM?
02D7 A7 00		STA A	0,X	SAVE CHAR
02D9 08		INX		
02DA 5A		DEC B		
02DB 2A F5		BPL	GETNA5	

02DD 0D		SEC	
02DE 39		RTS	RETURN
02DF 0C	GETNA6	CLC	
02E0 39		RTS	

* PRINT STRING WITHOUT CR & LF

02E1 A6 00	PDATA	LDA A 0, X	GET CHARACTER
02E3 27 08		BEQ PDATA2	NULL?
02E5 BD 71 12		JSR PUTCHR	PRINT IT
02E8 08		INX	BUMP TO NEXT
02E9 5A		DEC B	DEC COUNT
02EA 26 F5		BNE PDATA	
02EC 39		RTS	RETURN
02ED 5F	PDATA2	CLR B	
02EE 39		RTS	

* STRINGS

02EF 46	FILST	FCC	'FILES OF DRIVE NUMBER'
0305 04		FCB	4
0306 53	SYST	FCC	'SYNTAX ERROR'
0312 04		FCB	4
		END	FIL

NO ERROR(S) DETECTED

SYMBOL TABLE:

ALLFLG	010E	ASN	708C	BUFPNT	7094	CLASS	711B	COL	0117
COUNT	0116	DATPNT	0111	EOL	7082	ERR	010D	ERR2	01E0
EXIT	01CF	EXT	010B	FCB	7740	FIL	0100	FIL1	0119
FIL12	012A	FIL15	012C	FIL22	0132	FIL25	0144	FIL27	0149
FIL29	014C	FIL3	0156	FIL35	016E	FIL4	0174	FIL42	0197
FIL43	01A1	FIL45	01A4	FIL47	01B6	FIL5	01BD	FIL6	01C2
FILST	02EF	FMS	7806	FMSCLS	7803	GETCHR	710F	GETFIL	7127
GETHEX	713F	GETIR	0294	GETNA2	02B5	GETNA4	02CA	GETNAS	02D2
GETNA6	02DF	GETNAM	02AD	INDEX	010F	LOOP	0113	LSTTRM	7091
NAME	0103	NPL	0118	NXTCH	7121	OUTDEC	7133	OUTHEX	7139
PCRLF	711E	PDATA	02E1	PDATA2	02ED	POINT	0114	PRDI32	020B
PRDI33	021A	PRDI34	021D	PRDI35	0236	PRDI37	024E	PRDI45	027A
PRDI65	028E	PRDIR	01EE	PRDIR2	0200	PRDIR3	0203	PRDIR4	0254
PRDIR5	027F	PRDIR6	028A	PRDIR7	0293	PRNAM	029E	PRNAM2	02A8
PSTRNG	7118	PUTCHR	7112	RPTERR	713C	SERR	01D5	SPC	0020
SYST	0306	TSTTRM2	01ED	TSTTRM	01E3	VN	0102	WARMS	7103
WASN	708C	WIDTH	7084						

PRUL

The PRUL command is used to translate upper case only Text Processor files (those containing the 'cap' @ and ^ capitalization characters) into upper and lower case files. This is useful for converting old formatted text files.

DESCRIPTION

The general syntax of the PRUL command is:

```
PRUL,<input file spec>,<output file spec>
```

where the file specs default to TXT extensions and to the working drive. The name given to the output file must not already exist. The input file is read and all letters are converted to lower case unless preceded by an at-sign (@) or if surrounded by up-arrows (^). These are the same rules the Text Processor follows while in the capitalization mode. An example follows:

```
+++PRUL,BOOKREV,BOOKREVL
```

This command line would cause the file named BOOKREV.TXT to be read, the appropriate translation performed, and written back out to a file named BOOKREVL.TXT. The original file is left unchanged.

```

*
* TRANSLATE PR FILES TO LOWER CASE
*
*
* COPYRIGHT (C) 1978 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*

```

0100	ORG	\$0100	<i>\$A100</i>
------	-----	--------	---------------

* MAIN DISK EQUATES

7740	FCB	EQU	\$7740	<i>\$A040</i>
7806	FMS	EQU	\$7806	<i>\$B466</i>
7803	FMSCLS	EQU	\$7803	<i>\$B403</i>
7103	WARMIS	EQU	\$7103	<i>\$A003</i>
710F	GETCHR	EQU	\$710F	<i>\$A010</i>
7112	PUTCHR	EQU	\$7112	<i>\$A010</i>
7118	PSTRNG	EQU	\$7118	<i>\$A01E</i>
711E	PCRLF	EQU	\$711E	<i>\$A024</i>
7127	GETFIL	EQU	\$7127	<i>\$A02D</i>
712D	SETEXT	EQU	\$712D	<i>\$A033</i>
713C	RPTERR	EQU	\$713C	<i>\$A03F</i>
711B	CLASS	EQU	\$711B	<i>\$A021</i>

* MAIN STARTS HERE

0100 20 01	PRUL	BRA	PRUL1
------------	------	-----	-------

0102 01	VN	FCB	1
---------	----	-----	---

0103 CE 77 40	PRUL1	LDX	#FCB	
0106 BD 71 27		JSR	GETFIL	GET THE FILE NAME
0109 25 28		BCS	ERROR6	
010B 86 01		LDA A	#1	SET FOR READ
010D A7 00		STA A	0,X	SAVE IN FCB
010F BD 71 2D		JSR	SETEXT	SET DEFAULT EXT
0112 CE 77 40		LDX	#FCB	
0115 BD 78 06		JSR	FMS	CALL FMS
0118 27 25		BEQ	OPNROK	
011A A6 01	DSKERR	LDA A	1,X	GET ERROR TYPE
011C 81 03		CMP A	#3	FILE EXISTS?
011E 27 09		BEQ	ERROR2	
0120 81 04		CMP A	#4	NO FILE?
0122 27 0A		BEQ	ERROR4	
0124 BD 71 3C		JSR	RPTERR	REPORT ERROR
0127 20 10		BRA	DSKERR2	
0129 CE 01 FD	ERROR2	LDX	#FEST	POINT TO STRING

012C 29 08		BRA	ERROR7	POINT TO STRING
012E CE 01 DE	ERROR4	LDX	#NST	
0131 20 03		BRA	ERROR7	POINT TO STRING
0133 CE 01 EB	ERROR6	LDX	#ILST	PRINT STRING
0136 BD 71 18	ERROR7	JSR	PSTRNG	CLOSE ALL FILES
0139 BD 78 03	DSKER2	JSR	FMSCLS	RETURN TO FLEX
013C 7E 71 03		JMP	WARMs	POINT TO FCB
013F CE 02 09	OPNROK	LDX	#FCB2	GET FILE NAME
0142 BD 71 27		JSR	GETFIL	ERROR?
0145 25 EC		BCS	ERROR6	SET EXTENSION
0147 86 01		LDA A	#1	
0149 BD 71 2D		JSR	SETEXT	
014C CE 02 09		LDX	#FCB2	OPEN FOR WRITE
014F 86 02		LDA A	#2	
0151 A7 00		STA A	0, X	CALL FMS
0153 BD 78 06		JSR	FMS	ERROR?
0156 26 C2		BNE	DSKERR	POINT TO FCB
0158 CE 77 40	PRUL4	LDX	#FCB	READ CHARACTER
015B BD 78 06		JSR	FMS	ERROR?
015E 26 2E		BNE	PRUL8	IS IT ```
0160 81 5C		CMP A	#`\'	
0162 26 05		BNE	PRUL45	PROCESS BACKSLASH
0164 BD 01 C2		JSR	BKSL	
0167 20 EF		BRA	PRUL4	
0169 81 40	PRUL45	CMP A	#`@	IS IT `@`?
016B 27 2D		BEQ	ATS	
016D 81 5E		CMP A	#`↑	IS IT `↑`?
016F 27 33		BEQ	UPR	
0171 BD 71 1B		JSR	CLASS	CLASSIFY CHARACTER
0174 25 06		BCS	PRUL5	GRAPHICS?
0176 81 39		CMP A	#`9	IS IT NUMBER?
0178 23 02		BLS	PRUL5	
017A 8B 20		ADD A	#\$20	MAKE LOWER CASE
017C CE 02 09	PRUL5	LDX	#FCB2	POINT TO OUTPUT
017F BD 78 06		JSR	FMS	WRITE CHARACTER
0182 26 96		BNE	DSKERR	ERROR?
0184 20 D2		BRA	PRUL4	REPEAT
0186 CE 77 40	PRUL6	LDX	#FCB	POINT TO FCB
0189 BD 78 06		JSR	FMS	GET CHARACTER
018C 27 EE		BEQ	PRUL5	ERROR?
018E A6 01	PRUL8	LDA A	1, X	GET ERROR NUMBER
0190 81 08		CMP A	#8	IS IT EOF?
0192 26 47		BNE	DSER2	
0194 BD 78 03		JSR	FMSCLS	CLOSE ALL FILES
0197 7E 71 03		JMP	WARMs	RETURN TO FLEX

* PROCESS AT SIGN (@)

019A CE 77 40	ATS	LDX	#FCB	POINT TO FCB
019D BD 78 06		JSR	FMS	GET CHARACTER
01A0 26 EC		BNE	PRUL8	ERROR?
01A2 20 D8		BRA	PRUL5	

* PROCESS UP ARROW (↑)

BB TRANSLATE TO LOWER CASE

TSC MNEMONIC ASSEMBLER

PAGE 3

01A4	CE	77	40	UPA	LDX	#FCB	POINT TO FCB
01A7	BD	78	06		JSR	FMS	GET CHARACTER
01AA	26	E2			BNE	PRUL8	ERROR?
01AC	81	5C			CMP A	#'\\"	IS IT BACKSLASH?
01AE	26	04			BNE	UPA2	
01B0	8D	10			BSR	BKSL	PROCESS \
01B2	20	F0			BRA	UPA	REPEAT
01B4	81	5E		UPA2	CMP A	#'\t'	IS IT '\t'?
01B6	27	A0			BEQ	PRUL4	
01B8	CE	02	09		LDX	#FCB2	POINT TO OUTPUT
01BB	BD	78	06		JSR	FMS	WRITE CHARACTER
01BE	26	1B			BNE	DSER2	ERROR?
01C0	20	E2			BRA	UPA	REPEAT

* PROCESS BACKSLASH

01C2	CE	02	09	BKSL	LDX	#FCB2	POINT TO OUTPUT
01C5	BD	78	06		JSR	FMS	WRITE CHARACTER
01C8	26	11			BNE	DSER2	ERROR?
01CA	CE	77	40		LDX	#FCB	READ NEW CHARACTER
01CD	BD	78	06		JSR	FMS	
01D0	26	09			BNE	DSER2	ERROR?
01D2	CE	02	09		LDX	#FCB2	WRITE IT OUT
01D5	BD	78	06		JSR	FMS	
01D8	26	01			BNE	DSER2	ERROR?
01DA	39				RTS		RETURN
01DB	7E	01	1A	DSER2	JMP	DSKERR	

* STRINGS

01DE 4E	NSST	FCC	'NO SUCH FILE'	
01EA 04		FCB	4	
01EB 49	ILST	FCC	'ILLEGAL FILE NAME'	
01FC 04		FCB	4	
01FD 46	FEST	FCC	'FILE EXISTS'	
0208 04		FCB	4	
0209	FCB2	RMB	192	320
		END	PRIN	

NO ERROR(S) DETECTED

SYMBOL TABLE

ATS	019A	BKSL	01C2	CLASS	711B	DSER2	01DB	DSKER2	0139
DSKERR	011A	ERROR2	0129	ERPOR4	012E	ERROR6	0133	ERROR7	0136
FCB	7740	FCB2	0209	FEST	01FD	FMS	7806	FMSCLS	7803
GETCHR	710F	GETFIL	7127	ILST	01EB	NSST	01DE	OPNROK	013F
PCRLF	711E	PRUL	0100	PRUL1	0103	PRUL4	0158	PRUL45	0169
PRUL5	017C	PRUL6	0186	PRUL..	018E	PSTRNG	7118	PUTCHR	7112
PUTCHR	7120	PUTTME	7122	UDS	0121	UDS2	0124	UN	0122

DATE

The DATE utility allows the setting and displaying of the system date register. The date register is used by other FLEX utilities.

DESCRIPTION

The general syntax of the DATE command is:

```
DATE[,<date spec>]
```

where the date spec is in the form MM,DD,YY. If the date spec is left off the command line, the date will be displayed on the terminal. A few examples follow:

```
+++DATE  
+++DATE,10,6,78
```

The first example would display the current date on the terminal. The second example would set the date to 'October 6, 1978'. One of the FLEX utilities which make use of the system date register is the Text Processor.

```

*
* SET AND EXAMINE DATE
*
*
* COPYRIGHT (C) 1978 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*
*
```

* GLOBAL VARIABLES

7082	EOL	EQU	\$7082	\$A003
7740	FCB	EQU	\$7740	\$A810
708E	DATE	EQU	\$708E	\$AC1E
7091	LSTTRM	EQU	\$7091	\$AC11
7803	FMSCLS	EQU	\$7803	\$A463
7806	FMS	EQU	\$7806	\$A466
7103	WARMs	EQU	\$7103	\$AD63
710F	GETCHR	EQU	\$710F	\$AD15
7112	PUTCHR	EQU	\$7112	\$AD16
7118	PSTRNG	EQU	\$7118	\$AD1E
711E	PCRLF	EQU	\$711E	\$AD24
7133	OUTDEC	EQU	\$7133	\$AD39
713C	RPTERR	EQU	\$713C	\$AD3F
7121	NXTCH	EQU	\$7121	\$AD27
0100		ORG	\$0100	\$A100
0100 20 05	DATE0	BRA	DATE1	
0102 01	VN	FCB	1	VERSION NUMBER
0103 00 00	VALUE	FDB	0	
0105 00 00	DSNUM	FDB	0	
0107 B6 70 91	DATE1	LDA A	LSTTRM	CHECK TERM CHAR
010A 81 0D		CMP A	#\$D	IS IT CR?
010C 27 44		BEQ	PDAT	
010E B1 70 82		CMP A	EOL	IS IT EOL?
0111 27 3F		BEQ	PDAT	
0113 8D 2B		BSR	GETDAT	INPUT NUMBER
0115 25 20		BCS	DATE4	ERROR?
0117 81 0C		CMP A	#12	GREATER THAN 12?
0119 22 1C		BHI	DATE4	
011B B7 70 8E		STA A	DATE	SAVE MONTH
011E 8D 20		BSR	GETDAT	GET DAY NUMBER
0120 25 15		BCS	DATE4	ERROR?
0122 81 1F		CMP A	#31	GREATER THAN 31?
0124 22 11		BHI	DATE4	
0126 B7 70 8F		STA A	DATE+1	SAVE DAY
0129 8D 15		BSR	GETDAT	GET YEAR NUMBER

DATE UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 2

" 012B 25 0A	BCS	DATE4	ERROR?	
012D 81 63	CMP A	#99	GREATER THAN 99?	
012F 22 06	BHI	DATE4		
0131 B7 70 90	STA R	DATE+2	SAVE YEAR	
0134 7E 71 03	JMP	WARM\$	RETURN TO FLEX	
0137 CE 01 EC	DATE4	LDX	#SYST	POINT TO STRING
013A BD 71 18		JSR	PSTRNG	PRINT IT
013D 7E 71 03		JMP	WARM\$	RETURN TO FLEX

* INPUT DATE DIGIT

0140 BD 01 9D	GETDAT	JSR	INDEC	INPUT NUMBER
0143 25 0C		BCS	GETDA4	ERROR?
0145 5D		TST B		NUMBER THERE?
0146 27 08		BEQ	GETDA3	
0148 FF 01 03		STX	VALUE	SAVE VALUE
014B B6 01 04		LDA A	VALUE+1	GET LS PART
014E 0C		CLC		CLEAR ERRORS
014F 39		RTS		
0150 0D	GETDA3	SEC		SET ERRORS
0151 39	GETDA4	RTS		

* PRINT DATE

0152 BD 71 1E	PDAT	JSR	PCRLF	OUTPUT CR & LF
0155 B6 70 8E		LDA A	DATE	GET MONTH
0158 CE 01 F9		LDX	#MONTH	POINT TO TABLE
015B 4A	PDAT1	DEC A		CHECK DATE
015C 27 0E		BEQ	PDAT3	
015E 08	PDAT2	INX		FIND MONTH STRING
015F 6D 00		TST	0, X	
0161 26 FB		BNE	PDAT2	
0163 08		INX		
0164 20 F5		BRA	PDAT1	
0166 8D 2A	PDAT3	BSR	PST	GO PRINT IT
0168 86 20		LDA A	##20	OUTPUT SPACE
016A BD 71 12		JSR	PUTCHR	
016D 7F 01 03		CLR	VALUE	
0170 B6 70 8F		LDA A	DATE+1	GET DAY NUMBER
0173 B7 01 04		STA A	VALUE+1	
0176 CE 01 03		LDX	#VALUE	POINT TO IT
0179 5F		CLR B		CLEAR FLAG
017A BD 71 33		JSR	OUTDEC	PRINT DAY
017D CE 01 E7		LDX	#CST	POINT TO STRING
0180 8D 10		BSR	PST	PRINT IT
0182 B6 70 90		LDH A	DATE+2	GET YEAR
0185 B7 01 04		STA A	YVALUE+1	
0188 5F		CLR B		
0189 CE 01 03		LDX	#VALUE	POINT TO VALUE
018C BD 71 33		JSR	OUTDEC	PRINT YEAR
018F 7E 71 03		JMP	WARM\$	RETURN TO FLEX

* PRINT STRING

0192 A6 00	PST	LDA A	0, X	GET CHARACTER
0194 27 06		BEQ	PST2	IS IT NULL?
0196 BD 71 12		JSR	PUTCHR	OUTPUT CHARACTER
0199 08		INX		BUMP TO NEXT
019A 20 F6		BRA	PST	REPEAT
019C 39	PST2	RTS		

* INPUT DECIMAL NUMBER

019D 7F 01 05	INDEC	CLR	DSNUM	CLEAR VALUE
01A0 7F 01 06		CLR	DSNUM+1	
01A3 5F		CLR B		CLEAR COUNTER
01A4 BD 71 21	INDEC2	JSR	NXTCH	GET CHARACTER
01A7 25 33		BCS	INDEC4	FINISHED?
01A9 81 39		CMP A	#'9	IS IT NUMBER?
01AB 22 34		BHI	INDEC6	ERROR?
01AD 84 0F		AND A	#\$F	MASK VALUE
01AF 37		PSH B		SAVE COUNT
01B0 36		PSH A		SAVE NUMBER
01B1 B6 01 05		LDA A	DSNUM	GET NUMBER
01B4 F6 01 06		LDA B	DSNUM+1	
01B7 58		ASL B		MULT BY 6
01B8 49		ROL A		
01B9 58		ASL B		
01BA 49		ROL A		
01BB 58		ASL B		
01BC 49		ROL A		
01BD 78 01 06		ASL	DSNUM+1	TIMES 2
01C0 79 01 05		ROL	DSNUM	
01C3 FB 01 06		ADD B	DSNUM+1	ADD IN NEW
01C6 B9 01 05		ADC A	DSNUM	
01C9 F7 01 06		STA B	DSNUM+1	SAVE NEW
01CC 33		PUL B		
01CD FB 01 06		ADD B	DSNUM+1	
01D0 89 00		ADC A	#0	
01D2 F7 01 06		STA B	DSNUM+1	SAVE RESULT
01D5 B7 01 05		STA A	DSNUM	
01D8 33		PUL B		RESET COUNTER
01D9 5C		INC B		BUMP THE COUNT
01DA 20 C8		BRA	INDEC2	REPEAT
01DC FE 01 05	INDEC4	LDX	DSNUM	GET VALUE
01DF 0C		CLC		CLEAR ERRORS
01E0 39		RTS		
01E1 BD 71 21	INDEC6	JSR	NXTCH	GET NEXT CHAR
01E4 24 FB		BCC	INDEC6	TERM?
01E6 39		RTS		EXIT WITH ERROR

* TEXT STRINGS

01E7 2C	CST	FCC	" 19"
01E8 00		FCB	0
01EC 53	SYST	FCC	'SYNTAX ERROR'
01F8 04		FCB	4

* MONTH STRINGS

		MONTH	FCC	'JANUARY'
01F9	4A		FCB	0
0200	00		FCC	'FEBRUARY'
0201	46		FCB	0
0209	00		FCC	'MARCH'
020A	4D		FCB	0
020F	00		FCC	'APRIL'
0210	41		FCB	0
0215	00		FCC	'MAY'
0216	4D		FCB	0
0219	00		FCC	'JUNE'
021A	4A		FCB	0
021E	00		FCC	'JULY'
021F	4A		FCB	0
0223	00		FCC	'AUGUST'
0224	41		FCB	0
022A	00		FCC	'SEPTEMBER'
022B	53		FCB	0
0234	00		FCC	'OCTOBER'
0235	4F		FCB	0
023C	00		FCC	'NOVEMBER'
023D	4E		FCB	0
0245	00		FCC	'DECEMBER'
0246	44		FCB	0
024E	00		FCB	0

NO ERROR(S) DETECTED

SYMBOL TABLE:

RPT

The RPT command allows a command line to be repeatedly executed a specified number of times. This can be useful in diagnostic or demonstration applications.

DESCRIPTION

The general syntax of the RPT command is:

```
RPT,<repeat count>,<any command line>
```

where repeat count specifies the number of times the following command line should be executed. The command line may contain any FLEX utility except RPT and may also contain multiple commands by using the TTYSET End of Line character. An example follows:

```
+++RPT,6,LIST,BOOK4
```

This line would cause the file BOOK4 to be listed 6 times. Simple repeated demonstrations may be set up using the RPT command.

```

* REPEAT COMMAND UILITY <RPT>
*
*
* COPYRIGHT (C) 1978 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*

```

5F00		ORG	\$5F00	- 9F00
7103	WARMS	EQU	\$7103	7A03
7142	DOCMND	EQU	\$7142	7A04B
7094	BUFPNT	EQU	\$7094	7AC11
7121	NXTCH	EQU	\$7121	7AD27

* PROGRAM STARTS HERE

5F00 20 07	RPT	BRA	RPT1
------------	-----	-----	------

* TEMPORARY STORAGE

5F02 01	VN	FCB	1	
5F03 00 00	COUNT	FDB	0	
5F05 00 00	POINT	FDB	0	
5F07 00 00	DSNUM	FDB	0	
5F09 8D 29	RPT1	BSR	INDEC	GET RPT COUNT
5F08 25 24		BCS	RPT4	ERROR?
5F0D FF 5F 03		STX	COUNT	SAVE COUNT
5F10 FE 70 94		LDX	BUFPNT	GET BUFFER POINTER
5F13 FF 5F 05		STX	POINT	SAVE IT
5F16 B6 5F 04		LDA A	COUNT+1	CHECK IF COUNT=0
5F19 27 16		BEQ	RPT4	
5F1B BD 71 42	RPT2	JSR	DOCMND	GO DO COMMAND
5F1E 26 11		BNE	RPT4	ERROR?
5F20 B6 5F 04		LDA A	COUNT+1	GET COUNTER
5F23 4A		DEC A		SUBTRACT ONE
5F24 B7 5F 04		STA A	COUNT+1	STORE IT BACK
5F27 27 08		BEQ	RPT4	REPEAT?
5F29 FE 5F 05		LDX	POINT	RESET BUFFER POINTER
5F2C FF 70 94		STX	BUFPNT	
5F2F 20 EA		BRA	RPT2	REPEAT!
5F31 7E 71 03	RPT4	JMP	WARMS	RETURN TO FLEX

* INPUT DECIMAL NUMBER

5F34 7F 5F 07	INDEC	CLR	DSNUM	CLEAR VALUE
5F37 7F 5F 08		CLR	DSNUM+1	

REPEAT COMMAND UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 2

5F3A	5F		CLR B	CLEAR COUNTER
5F3B	BD	71 21	INDEC2	JSR NXTCH
5F3E	25	33		BCS INDEC4
5F40	81	39		CMP A #19
5F42	22	34		BHI INDEC6
5F44	84	0F		AND A #\$F
5F46	37			PSH B
5F47	36			PSH A
5F48	B6	5F 07		LDA A DSNUM
5F4B	F6	5F 08		LDA B DSNUM+1
5F4E	58			ASL B
5F4F	49			ROL A
5F50	58			ASL B
5F51	49			ROL A
5F52	58			ASL B
5F53	49			ROL A
5F54	78	5F 08		ASL DSNUM+1
5F57	79	5F 07		ROL DSNUM
5F5A	FB	5F 08		ADD B DSNUM+1
5F5D	B9	5F 07		RDC A DSNUM
5F60	F7	5F 08		STA B DSNUM+1
5F63	33			PUL B
5F64	FB	5F 08		ADD B DSNUM+1
5F67	89	00		ADC A #0
5F69	F7	5F 08		STA B DSNUM+1
5F6C	B7	5F 07		STA A DSNUM
5F6F	33			PUL B
5F70	5C			INC B
5F71	20	C8		BRA INDEC2
5F73	FE	5F 07	INDEC4	LDX DSNUM
5F76	0C			CLC
5F77	39			RTS
5F78	BD	71 21	INDEC6	JSR NXTCH
5F7B	24	FB		BCC INDEC6
5F7D	39			RTS
			END	RPT

NO ERROR(S) DETECTED

SYMBOL TABLE:

ECHO

The ECHO command is a very useful utility for incorporation into EXEC command files. It allows the echoing of ASCII strings to the terminal.

DESCRIPTION

The general syntax of the ECHO command is:

```
ECHO,<string>
```

where <string> is any string of printable characters terminated by a carriage return or end of line character. A few examples of the ECHO command follow:

```
+++ECHO,THE COPY PROCESS IS STARTING  
+++ECHO,TERMINAL 12
```

The first example would print the string "THE COPY PROCESS IS STARTING" on the terminal. The second example would print "TERMINAL 12". It is often useful to use ECHO in long EXEC command files to send instructive messages to the terminal to inform the operator of the status of the EXEC operation.

```

* ECHO STRING UTILITY
*
*
* COPYRIGHT (C) 1978 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*

```

* SYSTEM ROUTINE ADDRESSES

7103	WARMS	EQU	\$7103	7A003
7082	EOL	EQU	\$7082	7AC02
7112	PUTCHR	EQU	\$7112	7AD10
711E	PCRLF	EQU	\$711E	7A024
7121	NXTCH	EQU	\$7121	7AD27
7600		ORG	\$7600	7A100

* PROGRAM STARTS HERE

7600 20 01	ECHO	BRA	ECHO0	
7602 01	VN	FCB	1	VERSION NUMBER
7603 BD 71 1E	ECHO0	JSR	PCRLF	OUTPUT CR & LF
7606 BD 71 21	ECHO2	JSR	NXTCH	GET CHARACTER
7609 81 0D		CMP R	#\$D	IS IT CR?
760B 27 0A		BEQ	ECHO4	
760D B1 70 82		CMP R	EOL	IS T EOL CHAR?
7610 27 05		BEQ	ECHO4	
7612 BD 71 12		JSR	PUTCHR	OUTPUT CHARACTER
7615 20 EF		BRA	ECHO2	REPEAT
7617 7E 71 03	ECHO4	JMP	WARMS	RETURN TO FLEX
		END	ECHO	

NO ERROR(S) DETECTED

SYMBOL TABLE:

ECHO	7600	ECHO0	7603	ECHO2	7606	ECHO4	7617	EOL	7082
NXTCH	7121	PCRLF	711E	PUTCHR	7112	VN	7602	WARMS	7103

HECHO

The HECHO command is used for sending special character strings to the terminal. It is similar to the ECHO command, but HECHO allows control characters as well.

DESCRIPTION

The general syntax of the HECHO command is:

```
HECHO,<hex string>
```

where <hex string> is a list of hex digits representing ASCII characters. A few examples will demonstrate the use of HECHO.

```
+++HECHO,C  
+++HECHO,D,A,0,0,0  
+++HECHO,7,54,45,53,54,7
```

The first example will output a page eject (hex C) to the terminal. The next example will output a carriage return (hex D), a line feed (hex A), and then 4 null characters (hex 0). The last example will output an ASCII bell character (hex 7), then the string 'TEST', followed by another bell character.

```

*
* HEX STRING ECHO UTILITY
*
* COPYRIGHT 1978 BY
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*

```

* GLOBAL VARIABLES

7082	EQU	\$7082	<i>xA002</i>
7091	EQU	\$7091	<i>xA011</i>

* SYSTEM ROUTINE ADDRESSES

7103	WARM\$	EQU	\$7103	<i>xA003</i>
7112	PUTCHR	EQU	\$7112	<i>xA010</i>
711E	PCRLF	EQU	\$711E	<i>xA024</i>
713F	GETHEX	EQU	\$713F	<i>xA042</i>
7600		ORG	\$7600	<i>xA1C0</i>

7600 20 03	HECHO	BRA	HECHO0	
------------	-------	-----	--------	--

7602 01	VN	FCB	1	VERSION NUMBER
7603 00 00	VALUE	FDB	0	

7605 B6 70 91	HECHO0	LDA R	LSTTRM	CHECK TERM CHAR
7608 81 0D		CMP R	#\$D	IS IT CR?
760A 27 15		BEQ	HECHO2	
760C B1 70 82		CMP R	EOL	IS IT EOL CHAR?
760F 27 10		BEQ	HECHO2	
7611 BD 71 3F		JSR	GETHEX	GET HEX NUMBER
7614 25 0B		BCS	HECHO2	
7616 FF 76 03		STX	VALUE	SAVE VALUE
7619 B6 76 04		LDA A	VALUE+1	GET DIGIT
761C BD 71 12		JSR	PUTCHR	OUTPUT IT
761F 20 E4		BRA	HECHO0	REPEAT
7621 7E 71 03	HECHO2	JMP	WARM\$	RETURN TO FLEX
		END	HECHO	

NO ERROR(S) DETECTED

SYMBOL TABLE:

EOL	7082	GETHEX	713F	HECHO	7600	HECHO0	7605	HECHO2	7621
LSTTRM	7091	PCRLF	711E	PUTCHR	7112	VALUE	7603	VN	7602
WARM\$	7103								

FLIST

The FLIST utility is used to get a page formatted listing of a text type file. It is similar in operation to the LIST utility.

DESCRIPTION

The general syntax of the FLIST command is:

```
FLIST,<file spec>[,<line range>][,+<options>]
```

where the file spec designates the file to be listed and defaults to a TXT extension and to the working drive. The <line range> is the same as described in the LIST utility. If no range is specified, all lines will be displayed. Two options are supported, 'N' for line numbers, and 'P' for pagination. If the P option is specified, FLIST will prompt for a title. The title may contain a maximum of 40 characters. Each page will then be listed with a title and page number, followed by 54 lines of text (numbered if the N option was specified), and a hex \$0C formfeed character. A few examples will demonstrate the use of FLIST.

```
+--FLIST,CHAPTER1  
+--FLIST,LETTER,10-100,+NP  
+--FLIST,TEXT,50
```

The first example would cause the file named CHAPTER1.TXT to be displayed on the screen without line numbers or pagination. The second example would list LETTER.TXT from line number 10 through line 100 with line numbers and pagination. The last example would list the file named TEXT.TXT from line 50 to the end of the file. No line numbers will be output since the 'N' option was not specified.

```

*
* FORMATTED LIST UTILITY
*
*
* COPYRIGHT (C) 1978 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*

```

```
* GLOBAL VARIABLES
```

7082	EOL	EQU	\$7082	7AC02
7740	FCB	EQU	\$7740	7A840
7091	LSTTRM	EQU	\$7091	7AC11
7094	BUFPNT	EQU	\$7094	7AC14
7803	FMSCLS	EQU	\$7803	7B403
7806	FMS	EQU	\$7806	7B406
7103	WARMs	EQU	\$7103	7AD03
710F	GETCHR	EQU	\$710F	7AD15
7112	PUTCHR	EQU	\$7112	7AD18
7118	PSTRNG	EQU	\$7118	7AD1E
711B	CLASS	EQU	\$711B	7AD24
711E	PCRLF	EQU	\$711E	7AD24
7121	NXTCH	EQU	\$7121	7AD27
7127	GETFIL	EQU	\$7127	7AD2D
712D	SETEXT	EQU	\$712D	7AD33
7133	OUTDEC	EQU	\$7133	7AD34
713C	RPTERR	EQU	\$713C	7AD3F
70A3	OUTCHN	EQU	\$70A3	7AC
7139	OUTHEX	EQU	\$7139	7AD3C
7080	BSP	EQU	\$7080	7AC00
7081	DEL	EQU	\$7081	7AC01
0100		ORG	\$0100	7AC00

```
* PROGRAM STARTS HERE
```

0100 20 14	LIST	BRA	LIST1
------------	------	-----	-------

```
* TEMPORARY STORAGE
```

	VN	FCB	1	VERSION NUMBER
0102 01	TEMP	FDB	0	
0103 00 00	OUTF	FCB	0	
0105 00	NUMS	FCB	0	
0106 00	FIRST	FDB	0	
0107 00 00	LAST	FDB	\$FFFF	
0108 00 00	LINCNT	FDB	0	
010D 00	ZERO	FCB	0	

010E 0C	FF	FCB	\$C	
010F 00	PAGE	FCB	0	
0110 00	LINE	FCB	0	
0111 00 00	VALUE	FDB	0	
0113 00	PAGEN	FCB	0	
0114 00 00	DSNUM	FDB	0	

0116 BD 71 1E	LIST1	JSR	PCRLF	PRINT CR & LF
0119 CE 77 40		LDX	#FCB	POINT TO FCB
011C BD 71 27		JSR	GETFIL	GET FILE SPEC
011F 24 09		BCC	LIST15	ERROR?

* REPORT ILLEGAL NAME

0121 7E 02 5C	NAMER	JMP	ERROR6	REPORT ERROR
---------------	-------	-----	--------	--------------

* REPORT SYNTAX ERROR

0124 CE 03 84	SYNER	LDX	#SYNST	POINT TO STRING
0127 7E 02 5F		JMP	ERROR7	REPORT IT

* CONTINUE LIST

012A 86 01	LIST15	LDA A	#1	EXTENSION CODE
012C BD 71 2D		JSR	SETEXT	SET EXTENSION
012F BD 02 6B	LIST2	JSR	TSTTRM	TEST TERMINATOR
0132 27 4A		BEQ	LIST4	
0134 FE 70 94		LDX	BUFPNT	GET POINTER
0137 A6 00		LDA A	0,X	GET CHARACTER
0139 BD 71 1B		JSR	CLASS	CLASSIFY IT
013C 25 1C		BCS	LIST35	
013E BD 02 EE		JSR	INDEC	GET LINE NUMBER
0141 25 E1		BCS	SYNER	ERROR?
0143 FF 01 07		STX	FIRST	SAVE AS FIRST
0146 B6 70 91		LDA A	LSTTRM	CHECK TERM
0149 81 2D		CMP A	#`-`	IS IT `--`
014B 26 0D		BNE	LIST35	
014D BD 02 EE		JSR	INDEC	GET SECOND NUMBER
0150 25 D2		BCS	SYNER	
0152 FF 01 09		STX	LAST	SAVE AS LAST
0155 BD 02 6B	LIST3	JSR	TSTTRM	TEST TERMINATOR
0158 27 24		BEQ	LIST4	
015A 81 2B	LIST35	CMP A	#`+'	CHECK FOR OPTIONS
015C 27 07		BEQ	LIST37	
015E BD 71 21		JSR	NXTCH	GET NEXT CHAR
0161 24 C1		BCC	SYNER	TERM?
0163 20 F0		BRA	LIST3	
0165 BD 71 21	LIST37	JSR	NXTCH	GET NEXT CHARACTER
0168 25 EB		BCS	LIST3	TERM?
016A 84 5F		AND A	#\$5F	MAKE UPPER CASE
016C 81 4E		CMP A	#`N`	IS IT `N`?
016E 26 05		BNE	LIST38	
0170 B7 01 06		STA A	NUIS	SET NUMBERS OPT
0173 20 F0		BRA	LIST37	

FORMATTED LIST UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 3

0175 81 50		LIST38	CMP A #1P	IS IT 'P'?
0177 26 AB			BNE SYNER	
0179 B7 01 0F			STA A PAGE	SET PAGE OPT
017C 20 E7			BRA LIST37	
017E 7D 01 0F	LIST4		TST PAGE	PAGE ON?
0181 27 0B			BEQ LIST44	
0183 BD 02 76			JSR GETTL	INPUT TITLE
0186 7F 01 13			CLR PAGEN	SET PAGE NUM
0189 86 36			LDA A #54	
018B B7 01 10			STA A LINE	SET LINE COUNT
018E CE 77 40	LIST44		LDX #FCB	POINT TO FCB
0191 86 01			LDA A #1	OPEN FOR READ
0193 A7 00			STA A 0,X	
0195 BD 78 06			JSR FMS	
0198 27 03			BEQ LIST5	ERRORS?
019A 7E 02 43	LIST45		JMP LIST7	
019D 4F	LIST5		CLR A	BUMP LINE NUMBER
019E C6 01			LDA Z #1	
01A0 FB 01 0C			ADD B LINCNT+1	
01A3 F7 01 0C			STA B LINCNT+1	SAVE NEW VALUE
01A6 B9 01 0B			ADC A LINCNT	
01A9 B7 01 0B			STA A LINCNT	
01AC B1 01 09			CMP A LAST	END OF LIST?
01AF 25 07			BLO LIST52	
01B1 22 24			BHI LIST56	
01B3 F1 01 0A			CMP B LAST+1	
01B6 22 1F			BHI LIST56	
01B8 B1 01 07	LIST52		CMP A FIRST	AT FIRST LINE?
01B9 22 07			BHI LIST53	
01BD 25 08			BLO LIST54	
01BF F1 01 08			CMP B FIRST+1	
01C2 25 05			BLO LIST54	
01C4 86 01	LIST53		LDA A #1	SET OUTPUT FLAG
01C6 B7 01 05			STA A OUTF	
01C9 CE 77 40	LIST54		LDX #FCB	POINT TO FCB
01CC BD 78 06			JSR FMS	READ CHARACTER
01CF 27 0C			BEQ LIST6	ERRORS?
01D1 A6 01	LIST55		LDA A 1,X	GET ERROR TYPE
01D3 81 08			CMP A #8	IS IT EOF?
01D5 26 6C			BNE LIST7	
01D7 BD 78 03	LIST56		JSR FMSCLS	CLOSE ALL
01DA 7E 71 03			JMP WARMS	RETURN TO FLEX
01DD 7D 01 05	LIST6		TST OUTF	OUTPUT LINE?
01E0 27 39			BEQ LIST65	
01E2 36			PSH A	SAVE CHARACTER
01E3 7D 01 0F			TST PAGE	PAGING?
01E6 27 13			BEQ LIST63	
01E8 B6 01 10			LDA A LINE	CHECK LINE NUM
01EB 81 36			CMP A #54	
01ED 26 0C			BNE LIST63	
01EF 7C 01 13			INC PAGEN	BUMP PAGE NUMBER
01F2 B6 01 0E			LDA A FF	GET FORM FEED
01F5 BD 71 12			JSR PUTCHR	OUTPUT IT
01F8 BD 02 BB			JSR DOHDR	PRINT HEADER

FORMATTED LIST UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 4

01FB 32	LIST63	PUL A		RESTORE CHARACTER
01FC 7D 01 06		TST	NUMS	PRINT LINE NUM?
01FF 27 1A		BEQ	LIST65	
0201 36		PSH A		SAVE CHAR
0202 CE 01 0B		LDX	#LINCNT	POINT TO NUMBER
0205 C6 01		LDA B	#1	
0207 BD 71 33		JSR	OUTDEC	PRINT IT
020A 86 2E		LDA A	#'.'	OUTPUT '.'
020C BD 71 12		JSR	PUTCHR	
020F CE 01 0D		LDX	#ZERO	POINT TO ZEROES
0212 BD 71 39		JSR	OUTHEX	PRINT THEM
0215 86 3D		LDA A	#'=	PRINT '='
0217 BD 71 12		JSR	PUTCHR	
021A 32		PUL A		RESTORE CHAR
021B 81 0D	LIST65	CMP A	#\$D	IS IT CR?
021D 26 0E		BNE	LIST67	
021F 7D 01 05		TST	OUTF	DOING OUTPUT?
0222 27 06		BEQ	LIST66	
0224 7C 01 10		INC	LINE	BUMP LINE NUMBER
0227 BD 71 1E		JSR	PCRLF	DO CR & LF
022A 7E 01 9D	LIST66	JMP	LIST5	REPEAT
022D 81 0A	LIST67	CMP A	#\$0A	CHECK FOR LF
022F 27 08		BEQ	LIST68	
0231 7D 01 05		TST	OUTF	DOING OUTPUT?
0234 27 03		BEQ	LIST68	
0236 BD 71 12		JSR	PUTCHR	OUTPUT CHARACTER
0239 CE 77 40	LIST68	LDX	#FCB	POINT TO FCB
023C BD 78 06		JSR	FMS	GET CHARACTER
023F 27 DA		BEQ	LIST65	ERRORS?
0241 20 8E		BRA	LIST55	
0243 A6 01	LIST7	LDA A	1,X	GET ERROR TYPE
0245 81 03		CMP A	#3	FILE EXISTS?
0247 27 09		BEQ	ERROR2	
0249 81 04		CMP A	#4	NO FILE?
024B 27 0A		BEQ	ERROR4	
024D BD 71 3C		JSR	RPTERR	REPORT ERROR
0250 20 10		BRA	ERROR8	
0252 CE 03 78	ERROR2	LDX	#FEST	POINT TO STRING
0255 20 08		BRA	ERROR7	
0257 CE 03 59	ERROR4	LDX	#NSTT	POINT TO STRING
025A 20 03		BRA	ERROR7	
025C CE 03 66	ERROR6	LDX	#ILST	POINT TO STRING
025F BD 71 18	ERROR7	JSR	PSTRNG	PRINT STRING
0262 BD 78 03	ERROR8	JSR	FMSCLS	CLOSE ALL
0265 7E 71 03		JMP	WARM	
0268 7E 01 07		JMP	LIST56	

* TEST FOR TERMINATOR

026B B6 70 91	TSTTRM	LDA A	LSTTRM	GET LAST TERM
026E 81 0D		CMP A	#\$D	IS IT CR?
0270 27 03		BEQ	TSTTR2	
0272 B1 70 82		CMP A	EOL	IS IT EOL?
0275 39	TSTTR2	RTS		

* INPUT TITLE LINE

0276 CE 03 38	GETTL	LDX	#TTST	POINT TO STRING
0279 7C 70 A3		INC	OUTCHN	SET CHANNEL
027C BD 71 18		JSR	PSTRNG	PRINT STRING
027F CE 03 91		LDX	#TITLE	POINT TO BUFFER
0282 5F		CLR	B	
0283 BD 71 0F	GETTL2	JSR	GETCHR	INPUT TITLE
0286 B1 70 80		CMP	A	BACKSPACE?
0289 27 17		BEQ		
028B B1 70 81		CMP	A	DEL CHARACTER?
028E 27 E6		BEQ		
0290 81 0D		CMP	A	CR?
0292 27 15		BEQ		
0294 81 1F		CMP	A	CONTROL CHAR?
0296 23 EB		BLS		
0298 C1 27		CMP	B	BUFFER FULL?
029A 27 E7		BEQ		
029C A7 00		STA	A	SAVE CHARACTER
029E 08		INX		BUMP POINTER
029F 5C		INC	B	BUMP CHAR COUNT
02A0 20 E1		BRA		
02A2 5D	GETTL4	TST	B	TEST COUNT
02A3 27 D1		BEQ		
02A5 5A		DEC	B	DEC COUNT
02A6 09		DEX		DEC THE POINTER
02A7 20 DA		BRA		
02A9 86 20	GETTL6	LDA	A	SETUP SPACE
02AB A7 00		STA	A	PUT IN BUFFER
02AD 08		INX		BUMP TO NEXT
02AE 5C		INC	B	BUMP COUNTER
02AF C1 28		CMP	B	BUFFER FULL?
02B1 26 F6		BNE		
02B3 86 04		LDA	A	SET UP TERM
02B5 A7 00		STA	A	SAVE IN BUFFER
02B7 7F 70 A3		CLR		CLEAR CHANNEL
02BA 39		RTS		RETURN

* PRINT PAGE HEADER

02BB BD 71 1E	DOHDR	JSR	PCRLF	PRINT CR & LF
02BE BD 71 1E		JSR	PCRLF	
02C1 CE 03 91		LDX	#TITLE	POINT TO TITLE
02C4 BD 71 18		JSR	PSTRNG	PRINT IT
02C7 CE 03 40		LDX	#PGST	POINT TO STRING
02CA A6 00	DOHDR6	LDA	A	GET CHARACTER
02CC 27 06		BEQ	DOHDR7	IS IT NULL?
02CE BD 71 12		JSR	PUTCHR	PRINT IT
02D1 08		INX		BUMP TO NEXT
02D2 20 F6		BRA	DOHDR6	
02D4 5F	DOHDR7	CLR	B	
02D5 B6 01 13		LDA	A	GET PAGE NUM
02D8 B7 01 12		STA	A	SAVE NUMBER

02DB CE 01 11	LDX	#VALUE	POINT TO NUMBER
02DE BD 71 33	JSR	OUTDEC	PRINT IT
02E1 BD 71 1E	JSR	PCRLF	PRINT CR & LF
02E4 BD 71 1E	JSR	PCRLF	
02E7 BD 71 1E	JSR	PCRLF	
02EA 7F 01 10	CLR	LINE	CLEAR LINE COUNT
02ED 39	RTS		RETURN

* INPUT DECIMAL NUMBER

02EE 7F 01 14	INDEC	CLR	DSNUM	CLEAR VALUE
02F1 7F 01 15		CLR	DSNUM+1	
02F4 5F		CLR B		CLEAR COUNTER
02F5 BD 71 21	INDEC2	JSR	NXTCH	GET CHARACTER
02F8 25 33		BCS	INDEC4	FINISHED?
02FA 81 39		CMP A	#'9	IS IT NUMBER?
02FC 22 34		BHI	INDEC6	ERROR?
02FE 84 0F		AND A	#\$F	MASK VALUE
0300 37		PSH B		SAVE COUNT
0301 36		PSH A		SAVE NUMBER
0302 B6 01 14		LDA A	DSNUM	GET NUMBER
0305 F6 01 15		LDA B	DSNUM+1	
0308 58		ASL B		MULT BY 6
0309 49		ROL A		
030A 58		ASL B		
030B 49		ROL A		
030C 58		ASL B		
030D 49		ROL A		
030E 78 01 15		ASL	DSNUM+1	TIMES 2
0311 79 01 14		ROL	DSNUM	
0314 FB 01 15		ADD B	DSNUM+1	ADD IN NEW
0317 B9 01 14		ADC A	DSNUM	
031A F7 01 15		STA B	DSNUM+1	SAVE NEW
031D 33		PUL B		
031E FB 01 15		ADD B	DSNUM+1	
0321 89 00		ADC A	#0	
0323 F7 01 15		STA B	DSNUM+1	SAVE RESULT
0326 B7 01 14		STA A	DSNUM	
0329 33		PUL B		RESET COUNTER
032A 5C		INC B		BUMP THE COUNT
032B 20 C8		BRA	INDEC2	REPEAT
032D FE 01 14	INDEC4	LDX	DSNUM	GET VALUE
0330 0C		CLC		CLEAR ERRORS
0331 39		RTS		
0332 BD 71 21	INDEC6	JSR	NXTCH	GET NEXT CHAR
0335 24 FB		BCC	INDEC6	TERM?
0337 39		RTS		EXIT WITH ERROR
0338 54	TTST	FCC	'TITLE? '	
033F 04		FCB	4	
0340 20	PGST	FCC	'	PAGE 1.
0358 00		FCB	0	
0359 4E	NSST	FCC	'NO SUCH FILE'	
0365 04		FCB	4	

FORMATTED LIST UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 7

0366 49	ILST	FCC	'ILLEGAL FILE NAME'
0377 04		FCB	4
0378 46	FEST	FCC	'FILE EXISTS'
0383 04		FCB	4
0384 53	SYNST	FCC	'SYNTAX ERROR'
0390 04		FCB	4
0391	TITLE	RMB	41
		END	LIST

NO ERROR(S) DETECTED

SYMBOL TABLE:

PDEL

The PDEL command is a prompting delete utility. Either all files or only files matching a specified match list are displayed by name, one at a time, giving the option of deleting the file or keeping it. This command is very convenient for quickly removing a lot of no longer needed files from a disk.

DESCRIPTION

The general syntax of the PDEL command is:

```
PDEL[,<drive list>][,<match list>]
```

where drive list and match list are the same as described in the CAT command. Upon execution of PDEL, each file name will be printed at the terminal with the question of deleting it:

```
DELETE "FILE" ?
```

At this time three responses are valid. If a 'N' is typed, the file will be left intact and the next name will be displayed. If a 'Y' is typed, that file will be deleted. This utility DOES NOT ask if you are sure you want the file deleted, so make sure the first time! A carriage return may also be typed in response to the prompt at which time control will return back to FLEX. An example follows:

```
+++PDEL,1,.TXT
```

This command line would cause each file on drive 1 which has a TXT extension to be displayed and the delete option offered. Remember that once 'Y' has been typed to the prompt, that file is gone forever!

```

* PROMPT DELETE UTILITY PROGRAM
*
*
* COPYRIGHT (C) 1978 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*

```

```

* GLOBAL VARIABLES

```

7082	EOL	EQU	\$7082	✓AC02
7084	WIDTH	EQU	\$7084	✓AC04
7740	FCB	EQU	\$7740	✓AS40
708C	WASN	EQU	\$708C	✓AC0C
709C	ASN	EQU	WASN	
7091	LSTTRM	EQU	\$7091	✓AC11
7094	BUFPNT	EQU	\$7094	✓AC17
0020	SPC	EQU	\$20	
7803	FMSCLS	EQU	\$7803	✓A403
7806	FMS	EQU	\$7806	✓A4i6
7103	WARM5	EQU	\$7103	✓ADe3
710F	GETCHR	EQU	\$710F	✓ADf5
7112	PUTCHR	EQU	\$7112	✓ADi8
7118	PSTRNG	EQU	\$7118	✓AD1E
711B	CLASS	EQU	\$711B	✓ADz4
711E	PCRLF	EQU	\$711E	✓ADz4
7121	NXTCH	EQU	\$7121	✓ADL7
7127	GETFIL	EQU	\$7127	✓AD2H
7133	OUTDEC	EQU	\$7133	✓ADz4
7139	OUTHEX	EQU	\$7139	✓AD3C
713C	RPTERR	EQU	\$713C	✓AD3F
713F	GETHEX	EQU	\$713F	✓AD42

0100		ORG	\$0100	✓A100
------	--	-----	--------	-------

?

0100 20 18	PDE	BRA	PDE1
------------	-----	-----	------

0102 01	VN	FCB	1	VERSION NUMBER
0103 00	NAME	FCB	0, 0, 0, 0	
0107 00		FCB	0, 0, 0, 0	
0108 00	EXT	FCB	0, 0, 0	
010E 00	ALLFLG	FCB	0	
010F 00	LOOP	FCB	0	
0110 00 00	POINT	FDB	0	
0112 00 00	TEMP	FDB	0	
0114 00 00	TEMP2	FDB	0	
0116 00 00	INDEX	FDB	0	
0118 00 00	DATPNT	FDB	0	

* PROGRAM HERE

011A FE 70 94	PDE1	LDX	BUFPNT	POINT TO BUFFER
011D FF 01 10		STX	POINT	SAVE POS
0120 FE 01 10	PDE22	LDX	POINT	GET POINTER
0123 A6 00		LDA A	0,X	GET CHARACTER
0125 81 2E		CMP A	#1.	IS IT PERIOD?
0127 27 09		BEQ	PDE25	
0129 BD 71 1B		JSR	CLASS	CLASSIFY IT
012C 25 04		BCS	PDE25	TERM?
012E 81 39		CMP A	#19	IS IT NUMBER?
0130 23 12		BLS	PDE3	
0132 7D 01 0F	PDE25	TST	LOOP	LOOPING?
0135 27 03		BEQ	PDE29	
0137 7E 71 03	PDE27	JMP	WARM	RETURN TO FLEX
013A CE 77 40	PDE29	LDX	#FCB	POINT TO FCB
013D B6 70 8C		LDA A	ASN	GET DRIVE
0140 A7 03		STA A	3,X	SET IN FCB
0142 20 1E		BRA	PDE4	
0144 FE 01 10	PDE3	LDX	POINT	RESTORE POINTER
0147 FF 70 94		STX	BUFPNT	
014A BD 71 21		JSR	NXTCH	GET NEXT CHAR
014D 84 03		RND A	#3	MASK NUMBER
014F CE 77 40		LDX	#FCB	POINT TO FCB
0152 A7 03		STA A	3,X	SAVE IN DRIVE
0154 BD 71 21		JSR	NXTCH	GET NEXT CHAR
0157 25 03		BCS	PDE35	TERM?
0159 7E 01 A3		JMP	SERR	IF NOT, ERROR
015C FE 70 94	PDE35	LDX	BUFPNT	SET POINTER
015F FF 01 10		STX	POINT	
0162 BD 71 1E	PDE4	JSR	PCRLF	PRINT CR & LF
0165 BD 01 CD	PDE42	JSR	TSTTRM	TEST TERMINATOR
0168 26 08		BNE	PDE45	
016A 7F 01 0E		CLR	ALLFLG	CLEAR FLAG
016D 20 21		BRA	PDE6	
016F 7E 01 A8	PDE43	JMP	ERR	REPORT ERROR
0172 FE 70 94	PDE45	LDX	BUFPNT	RESET POINTER
0175 A6 00		LDA A	0,X	GET NEXT CHAR
0177 81 2E		CMP A	#1.	IS IT PERIOD?
0179 27 10		BEQ	PDE5	
017B BD 71 1B		JSR	CLASS	CLASSIFY IT
017E 25 23		BCS	SERR	TERM?
0180 81 39		CMP A	#19	IS IT NUMBER?
0182 22 07		BHI	PDE5	
0184 BD 71 21	PDE47	JSR	NXTCH	GET NEXT
0187 24 FB		BCC	PDE47	TERM?
0189 20 DA		BRA	PDE42	
019B BD 02 C0	PDE5	JSR	GETNAM	INPUT NAME
019E 25 13		BCS	SERR	ERROR?
0190 BD 01 D9	PDE6	JSR	DODLR	GO DO DELETE
0193 BD 38		BSR	TSTTRM	TEST TERM
0195 26 F4		BNE	PDE5	
0197 BD 71 1E		JSR	PCRLF	PRINT CR & LF
019A 7E 01 20		JMP	PDE22	REPERT

* EXIT DIR UTILITY

019D BD 71 1E	EXIT	JSR	PCRLF	PRINT CR & LF
01A0 7E 71 03		JMP	WARMs	RETURN TO FLEX

* REPORT SYNTAX ERROR

01A3 CE 03 3B	SERR	LDX	#SYNST	POINT TO STRING
01A6 20 1C		BRA	ERROR7	

* REPORT DISK ERROR

01A8 A6 01	ERR	LDA A	1,X	GET ERROR TYPE
01AA 81 03		CMP A	#3	FILE EXISTS?
01AC 27 09		BEQ	ERROR2	
01AE 81 04		CMP A	#4	NO FILE?
01B0 27 0A		BEQ	ERROR4	
01B2 BD 71 3C		JSR	RPTERR	REPORT ERROR
01B5 20 18		BRA	ERROR8	
01B7 CE 03 2E	ERROR2	LDX	#FEST	POINT TO STRING
01BA 20 08		BRA	ERROR7	
01BC CE 03 0F	ERROR4	LDX	#NSST	POINT TO STRING
01BF 20 03		BRA	ERROR7	
01C1 CE 03 1C	ERROR6	LDX	#ILST	POINT TO STRING
01C4 BD 71 18	ERROR7	JSR	PSTRNG	PRINT STRING
01C7 BD 78 03	ERROR8	JSR	FMSCLS	CLOSE ALL
01CA 7E 71 03	ERR2	JMP	WARMs	RETURN TO FLEX

* TEST FOR TERMINATOR

01CD B6 70 91	TSTTRM	LDA A	LSTTRM	CHECK LAST TERM
01D0 81 0D		CMP A	#\$D	IS IT CR?
01D2 27 03		BEQ	TSTTR2	
01D4 B1 70 82		CMP A	EOL	IS IT EOL?
01D7 39	TSTTR2	RTS		

* DO DELETE PROCESS

01D8 CE 77 40	DODLR	LDX	#FCB	POINT TO FCB
01DB 86 06		LDA A	#6	OPEN DIRECTORY
01DD A7 00		STA A	0,X	
01DF BD 72 06		JSR	FMS	
01E2 26 C4		BNE	ERR	ERRORS?
01E4 7C 01 0F		INC	LOOP	SET LOOP FLAG
01E7 BD 71 1E	DODLR2	JSR	PCRLF	PRINT CR & LF
01EA BD 02 AA	DODLR3	JSR	GETIR	GET INFO RECORD
01ED 27 03		BEQ	DODL32	ERRORS?
01EF 7E 02 A0		JMP	DODLR6	
01F2 CE 77 40	DODL32	LDX	#FCB	POINT TO FCB
01F5 6D 04		TST	4,X	NAME PRESENT?
01F7 26 03		BNE	DODL34	
01F9 7E 02 A9		JMP	DODLR7	
01FC 2B EC	DODL34	BMI	DODLR3	DELETED NAME?

PROMPT DELETE UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 4

01FE CE 77 44	LDX	#FCB+4	POINT TO NAME
0201 7D 01 0E	TST	ALLFLG	ALL FILES?
0204 26 27	BNE	DODLR4	
0206 FF 01 18	STX	DATPNT	SAVE POINTER
0209 CE 01 03	LDX	#NAME	TRY TO MATCH NAME
020C FF 01 16	STX	INDEX	
020F FE 01 16	LDX	INDEX	
0212 8C 01 0E	CPX	#EXT+3	
0215 27 16	BEQ	DODLR4	
0217 A6 00	LDA A	0,X	
0219 08	INX		
021A FF 01 16	STX	INDEX	
021D FE 01 18	LDX	DATPNT	
0220 4D	TST A		CHAR NULL?
0221 27 04	BEQ	DODL37	
0223 A1 00	CMP A	0,X	DO COMPARISON
0225 26 C3	BNE	DODLR3	
0227 08	DODL37	INX	MOVE TO NEXT
0228 FF 01 18	STX	DATPNT	
022B 20 E2	BRA	DODL35	
022D CE 77 43	DODLR4	LDX	POINT TO FCB
0230 FF 01 12	STX	TEMP	SAVE
0233 CE 03 48	LDX	#FCB2+3	POINT TO FCB 2
0236 FF 01 14	STX	TEMP2	
0239 C6 0C	LDA B	#12	SET COUNTER
023B FE 01 12	DODL45	LDX	GET POINTER
023E A6 00	LDA A	0,X	GET CHARACTER
0240 08	INX		BUMP TO NEXT
0241 FF 01 12	STX	TEMP	
0244 FE 01 14	LDX	TEMP2	RESET POINTER
0247 A7 00	STA A	0,X	PUT CHARACTER
0249 08	INX		
024A FF 01 14	STX	TEMP2	
024D 5A	DEC B		DEC THE COUNT
024E 26 EB	BNE	DODL45	
0250 CE 03 02	LDX	#DST	POINT TO STRING
0253 BD 71 18	JSR	PSTRNG	PRINT IT
0256 CE 03 49	LDX	#FCB2+2	POINT TO DRIVE
0259 6F 00	CLR	0,X	
025B 5F	CLR B		
025C BD 71 33	JSR	OUTDEC	PRINT DRIVE
025F 86 2E	LDA A	#1.	PRINT ". "
0261 BD 71 12	JSR	PUTCHR	
0264 CE 03 4B	LDX	#FCB2+4	POINT TO NAME
0267 C6 08	LDA B	#8	SET COUNTER
0269 BD 02 B4	JSP	PRNAM	PRINT IT
026C 86 2E	LDA A	#1.	PRINT "
026E BD 71 12	JSR	PUTCHR	"
0271 C6 03	LDA B	#3	SET COUNTER
0273 BD 02 B4	JSR	PRNAM	PRINT EXT
0276 CE 03 0B	LDX	#DST2	POINT TO STRING
0279 CG 03	LDA B	#3	SET COUNT
027B BD 02 F4	JSR	PLATA	PRINT IT
027E BD 71 0F	JSR	GITCHR	GET RESPONSE

PROMPT DELETE UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 5

0281 81 0D	CMP A	#\$D	IS IT CR?
0283 26 06	BNE	DODL48	
0285 BD 71 1E	JSR	PCRLF	PRINT CR & LF
0288 7E 71 03	JMP	WARM\$	RETURN TO FLEX
028B 84 5F	AND A	##\$5F	MASK TO UPPER
028D 81 59	CMP A	#'Y	IS IT "Y"?
028F 26 0C	BNE	DODLR5	
0291 CE 03 47	LDX	#FCB2	POINT TO FCB
0294 86 0C	LDA A	#12	DELETE CODE
0296 A7 00	STA A	0,X	
0298 BD 78 06	JSR	FMS	DELETE FILE
029B 27 00	BEQ	DODLR5	ERRORS?
029D 7E 01 EA	JMP	DODLR3	REPEAT
02A0 A6 01	DODLR6	LDA A	GET ERROR TYPE
02A2 81 08		CMP A	IS IT EOF?
02A4 27 03	DODL65	BEQ	DODLR7
02A6 7E 01 A8		JMP	REPORT ERROR
02A9 39	DODLR7	RTS	

* GET INFO RECORD

02AA CE 77 40	GETIR	LDX	#FCB	POINT TO FCB
02AD 86 07		LDA A	#7	GET IR CODE
02AF A7 00		STA A	0,X	
02B1 7E 78 06		JMP	FMS	CALL FMS

* PRINT FILE NAME

02B4 A6 00	PRNAM	LDA A	0,X	GET CHARACTER
02B6 27 03		BEQ	PRNAM2	
02B8 BD 71 12		JSR	PUTCHR	OUTPUT CHAR
02B8 08	PRNAM2	INX		BUMP TO NEXT
02BC 5A		DEC B		
02BD 26 F5		BNE	PPNAM	
02BF 39		RTS		RETURN

* GET FILE MATCH NAME

02C0 7F 01 0E	GETNAM	CLR	ALLFLG	CLEAR ALL
02C3 CE 01 03		LDX	#NAME	POINT TO BUFFER
02C6 C6 08		LDA B	#11	SET COUNT
02C8 6F 00	GETNA2	CLR	0,X	CLEAR BUFFER
02CA 08		INX		
02CB 5A		DEC B		
02CC 26 FA		BNE	GETNA2	
02CE FE 70 94		LDX	BUFFPNT	POINT TO NAME
02D1 A6 00		LDA A	0,X	GET CHARACTER
02D3 81 2E		CMP A	#'	IS IT '.'?
02D5 27 06		BEQ	GETNR4	
02D7 CE 00 FF		LDX	#NAME-4	RESET POINTER
02D9 7E 71 27		JMP	GETFIL	GET FILE NAME
02DD 7C 70 95	GETNR4	INC	BUFFPNT+1	FIX POINTER
02E0 C6 03		LDA B	#3	SET COUNT
02E2 CE 01 08		LDX	#EXT	POINT TO EXT

PROMPT DELETE UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 6

02E5 BD 71 21	GETNA5	JSR	NXTCH	GET CHARACTER
02E8 25 08		BCS	GETNA6	TERM?
02EA A7 00		STA A	0,X	SAVE CHAR
02EC 08		INX		
02ED 5A		DEC B		
02EE 2A F5		BPL	GETNA5	
02F0 0D		SEC		
02F1 39		RTS		RETURN
02F2 0C	GETNA6	CLC		
02F3 39		RTS		

* PRINT STRING WITHOUT CR & LF

02F4 A6 00	PDATA	LDA R	0,X	GET CHARACTER
02F6 27 08		BEQ	PDATA2	NULL?
02F8 BD 71 12		JSR	PUTCHR	PRINT IT
02FB 08		INX		BUMP TO NEXT
02FC 5A		DEC B		DEC COUNT
02FD 26 F5		BNE	PDATA	
02FF 39		RTS		RETURN
0300 5F	PDATA2	CLR B		
0301 39		RTS		

* STRINGS

0302 44	DST	FCC	'DELETE "	
030A 04		FCB	4	
030B 22	DST2	FCC	'"? '	
030E 04		FCB	4	
030F 4E	NSST	FCC	'NO SUCH FILE'	
031B 04		FCB	4	
031C 49	ILST	FCC	'ILLEGAL FILE NAME'	
032D 04		FCB	4	
032E 46	FEST	FCC	'FILE EXISTS'	
0339 04		FCB	4	
033A 53	SYNST	FCC	'SYNTAX ERROR'	
0346 04		FCB	4	
0347	FCB2	RMB	192 .3.20	
		END	PDE	

NO ERROR(S) DETECTED

15(10000.0000
61	1...000
1 4	0 1

SLEEP

The SLEEP command is used to eat up time. It is handy to use in special applications in a EXEC command file.

DESCRIPTION

The general syntax of the SLEEP command is:

SLEEP,N

where N is the number of seconds the system is to "sleep". It should be noted that large values of N (greater than 15) may not timeout exactly. Once the SLEEP command is executed, it cannot be interrupted. You must wait for the entire time period specified. An example will demonstrate SLEEP's use.

+++SLEEP,10

This command line would cause the system to lockup or sleep for approximately 10 seconds.

```

*
* SLEEP UTILITY
*
*
* COPYRIGHT (C) 1978 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*

```

7600 ORG \$7600 *#A100*

* EQUATES

7103	WARMS	EQU	\$7103	<i>#A003</i>
7121	NXTCH	EQU	\$7121	<i>#A027</i>

* PROGRAM STARTS HERE

7600 20 05 SLEEP BRA SLEEP1

7602 01	YN	FCB	1	
7603 00 00	VALUE	FDB	0	
7605 00 00	DSNUM	FDB	0	
7607 8D 1A	SLEEP1	BSR	INDEC	GET COUNT
7609 FF 76 03	SLEEP4	STX	VALUE	SAVE COUNT
760C 27 12		REQ	SLEEP6	
760E CE 00 00		LDX	#0	SET COUNTER
7611 09	LOOP1	DEX		DEC THE COUNT
7612 26 FD		BNE	LOOP1	
7614 CE 00 00		LDX	#0	SET COUNT
7617 09	LOOP2	DEX		DEC THE COUNT
7618 26 FD		BNE	LOOP2	
761A FE 76 03		LDX	VALUE	GET VALUE
761D 09		DEX		DEC THE VALUE
761E 20 E9		BRA	SLEEP4	REPEAT PROCESS
7620 7E 71 03	SLEEP6	JMP	WARMS	RETURN TO FLEX

* INPUT DECIMAL NUMBER

7623 7F 76 05	INDEC	CLR	DSNUM	CLEAR VALUE
7626 7F 76 06		CLR	DSNUM+1	
7629 5F		CLR B		CLEAR COUNTER
762A BD 71 21	INDEC2	JSR	NXTCH	GET CHARACTER
762D 25 33		BCS	INDEC4	FINISHED?
762F 81 39		CMP A	#19	IS IT NUMBER?
7631 22 34		BHI	INDEC6	ERROR?
7633 84 0F		AND A	#1F	MASK VALUE
7635 37		PSH B		SAVE COUNT
7636 36		PSH A		SAVE NUMBER

SLEEP UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 2

7637 B6 76 05	LDA A	DSNUM	GET NUMBER
763A F6 76 06	LDA B	DSNUM+1	
763D 58	ASL B		MULT BY 6
763E 49	ROL A		
763F 58	ASL B		
7640 49	ROL A		
7641 58	ASL B		
7642 49	ROL A		
7643 78 76 06	ASL	DSNUM+1	TIMES 2
7646 79 76 05	ROL	DSNUM	
7649 FB 76 06	ADD B	DSNUM+1	ADD IN NEW
764C B9 76 05	ADC A	DSNUM	
764F F7 76 06	STA B	DSNUM+1	SAVE NEW
7652 33	PUL B		
7653 FB 76 06	ADD B	DSNUM+1	
7656 89 00	ADC A	#0	
7658 F7 76 06	STA B	DSNUM+1	SAVE RESULT
765B B7 76 05	STA A	DSNUM	
765E 33	PUL B		RESET COUNTER
765F 5C	INC B		BUMP THE COUNT
7660 20 C8	BRA	INDEC2	REPEAT
7662 FE 76 05	INDEC4	LDX DSNUM .	GET VALUE
7665 0C	CLC		CLEAR ERRORS
7666 39	RTS		
7667 BD 71 21	INDEC6	JSR NXTCH	GET NEXT CHAR
766A 24 FB		BCC INDEC6	TERM?
766C 39	RTS		EXIT WITH ERROR
	END	SLEEP	

NO ERROR(S) DETECTED

SYMBOL TABLE:

DSNUM	7605	INDEC	7623	INDEC2	762A	INDEC4	7662	INDEC6	7667
LOOP1	7611	LOOP2	7617	NXTCH	7121	SLEEP	7608	SLEEP1	7607
SLEEP4	7609	SLEEP6	7620	VALUE	7603	VN	7602	WARMS	7103

REMSPC

The REMSPC command will remove all excess spaces from a text file. This is a useful utility for file conversions and file space reduction.

DESCRIPTION

The general syntax of the REMSPC command is:

```
REMSPC,<input file spec>,<output file spec>
```

where the file specs default to a TXT extension and to the working drive. The input file is the file to be processed and the output file name must not already exist on the disk. REMSPC will convert all occurrences of two or more spaces into a single space unless the line starts with an asterisk (*) in column 1 (a comment line). These comment lines are passed unmodified to the output file. An example of using REMSPC follows:

```
+++REMSPC,SOURCE,SOURCE2
```

This would cause the file named SOURCE.TXT to be read from the working drive, all excess spaces removed, and written back to a file named SOURCE2.TXT on the working drive.

```

*
* REMOVE SPACES UTILITY
*
*
* COPYRIGHT (C) 1978 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*

```

* EQUATES

7740	RFCB	EQU	\$7740	JA240
7127	GETFIL	EQU	\$7127	JA120
712D	SETEXT	EQU	\$712D	JA133
7806	FMS	EQU	\$7806	JA401
7803	FMSCLS	EQU	\$7803	JA403
713C	RPTERR	EQU	\$713C	JA13F
7103	WARMIS	EQU	\$7103	JA003
7118	PSTRNG	EQU	\$7118	JA1E
0100		ORG	\$0100	JA100

* PROGRAM STARTS HERE

0100 20 02	XL	BRA	XL1
------------	----	-----	-----

* TEMPORARY STORAGE

0102 01	VN	FCB	1	
0103 00	SPCFLG	FCB	0	
0104 CE 77 40	XL1	LDX	#RFCB	POINT TO FCB
0107 BD 71 27		JSR	GETFIL	GET FILE NAME
010A 25 74		BCS	ERROR6	ERROR?
010C 86 01		LDR A	#1	SET EXTENSION
010E A7 00		STA A	0,X	
0110 BD 71 2D		JSR	SETEXT	
0113 CE 77 40		LDX	#WFCB	
0116 BD 78 06		JSR	FMS	OPEN FOR READ
0119 26 4C		BNE	XL5	ERROR?
011B CE 01 EF		LDX	#WFCB	POINT TO OUTPUT FCB
011E BD 71 27		JSR	GETFIL	GET FILE NAME
0121 25 5D		BCS	ERROR6	ERROR?
0123 86 01		LDA A	#1	SET EXTENSION
0125 BD 71 2D		JSR	SETEXT	
0128 CE 01 EF		LDX	#WFCB	
012B 86 02		LDA A	#2	SET FOR WRITE
012D A7 00		STA A	0,X	
012F BD 78 06		JSR	FMS	OPEN FILE
0132 26 33		BNE	XL5	ERROR?

REMOVE SPACES UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 2

0134 7F 01 03	XL2	CLR	SPCFLG	CLEAR SPACE FLAG
0137 CE 77 40		LDX	#RFCB	POINT TO INPUT
013A BD 78 06		JSR	FMS	GET CHARACTER
013D 26 5E		BNE	XL75	ERROR?
013F 81 2A		CMP A	#'*	IS IT COMMENT?
0141 27 62		BEQ	XL8	
0143 81 20	XL3	CMP A	#\$20	IS IT SPACE?
0145 26 05		BNE	XL4	
0147 B7 01 03		STA A	SPCFLG	SET FLAG
014A 20 49		BRA	XL7	
014C 81 0D	XL4	CMP A	#\$D	IS IT CR?
014E 27 69		BEQ	XL9	
0150 81 1F		CMP A	#\$1F	IS IT CONTROL?
0152 23 41		BLS	XL7	
0154 7D 01 03		TST	SPCFLG	FLAG SET?
0157 27 34		BEQ	XL65	
0159 7F 01 03		CLR	SPCFLG	CLEAR THE FLAG
015C 36		PSH A		SAVE THE CHARACTER
015D 86 20		LDA A	#\$20	SETUP SPACE
015F CE 01 EF		LDX	#WFCB	POINT TO OUT
0162 BD 78 06		JSR	FMS	WRITE SPACE
0165 27 25		BEQ	XL6	ERROR?
0167 A6 01	XL5	LDA A	1,X	GET ERROR TYPE
0169 81 03		CMP A	#3	FILE EXISTS?
016B 27 09		BEQ	ERROR2	
016D 81 04		CMP A	#4	NO FILE?
016F 27 0A		BEQ	ERROR4	
0171 BD 71 3C		JSR	RPTERR	REPORT ERROR
0174 20 10		BRA	XL55	
0176 CE 01 E3	ERROR2	LDX	#FEST	POINT TO STRING
0179 20 08		BRA	ERROR7	
017B CE 01 C4	ERROR4	LDX	#NST	POINT TO STRING
017E 20 03		BRA	ERROR7	
0180 CE 01 D1	ERROR6	LDX	#ILST	POINT TO STRING
0183 BD 71 18	ERROR7	JSR	PSTRNG	PRINT STRING
0186 BD 78 03	XL55	JSR	FMSCLS	CLOSE ALL FILES
0189 7E 71 03	XL57	JMP	WARM	RETURN TO FLEX
018C 32	XL6	PUL A		RESTORE CHARACTER
018D CE 01 EF	XL65	LDX	#WFCB	POINT TO OUTPUT
0190 BD 78 06		JSR	FMS	WRITE CHARACTER
0193 26 D2		BNE	XL5	ERROR?
0195 CE 77 40	XL7	LDX	#RFCB	POINT TO INPUT
0198 BD 78 06		JSR	FMS	READ CHARACTER
019B 27 A6		BEQ	XL3	ERROR?
019D A6 01	XL75	LDA A	1,X	GET ERROR NUMBER
019F 81 08		CMP A	#8	IS IT EOF?
01A1 26 C4		BNE	XL5	
01A3 20 E1		BRA	XL55	
01A5 CE 61 EF	XL8	LDX	#WFCB	PROCESS COMMENT
01A8 BD 78 06		JSR	FMS	WRITE CHARACTER
01AB 26 BA		BNE	XL5	ERROR?
01AD CE 77 40		LDX	#RFCB	GET NEXT CHARACTER
01B0 BD 78 06		JSR	FMS	
01B3 26 E8		BNE	XL75	ERROR?

REMOVE SPACES UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 3

01B5 81 0D	CMP A	#\$D	IS IT CR?
01B7 26 EC	BNE	XL8	
01B9 CE 01 EF XL9	LDX	#WFCB	WRITE-CR OUT
01BC BD 78 06	JSR	FMS	
01BF 26 R6	BNE	XL5	ERROR?
01C1 7E 01 34	JMP	XL2	REPEAT

* STRINGS

01C4 4E	NSST	FCC	'NO SUCH FILE'
01D0 04		FCB	4
01D1 49	ILST	FCC	'ILLEGAL FILE NAME'
01E2 04		FCB	4
01E3 46	FEST	FCC	'FILE EXISTS'
01EE 04		FCB	4
01EF	WFCB	RMB	192 <i>320</i>
		END	XL

NO ERROR(S) DETECTED

SYMBOL TABLE:

ERROR2 0176	ERROR4 017B	ERROR6 0180	ERROR7 0183	FEST 01E3
FMS 7806	FMSCLS 7803	GETFIL 7127	ILST 01D1	NSST 01C4
PSTRNG 7118	RFCB 7740	RPTERR 7130	SETEXT 712D	SPCFLG 0103
VN 0102	WARMS 7103	WFCB 01EF	XL 0100	XL1 0104
XL2 0134	XL3 0143	XL4 014C	XL5 0167	XL55 0186
XL57 0189	XL6 019C	XL65 018D	XL7 0195	XL75 019D
XL8 01A5	XL9 01B9			

CONCAT

The CONCAT command allows the listing and concatenation of several files. The files will be listed to the output device (file if the O command is used), one after the other.

DESCRIPTION

The general syntax of the CONCAT command is:

```
CONCAT,<file spec list>
```

where the file spec list is a list of file specifications separated by commas. The file specs will default to a TXT extension and to the working drive. An example follows:

```
+--CONCAT,CHAPT1,CHAPT2,CHAPT3,CHAPT4
```

This will display the contents of the files CHAPT1.TXT, CHAPT2.TXT, CHAPT3.TXT, and CHAPT4.TXT, one after the other on the terminal.

aanleenschakelen

```

* CONCATENATE FILES UTILITY
*
* COPYRIGHT (C) 1978 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
* MAIN DISK EQUATES

```

7740	FCB	EQU	\$7740	JA340
7806	FMS	EQU	\$7806	JA406
7803	FMSCLS	EQU	\$7803	JA403
7103	WARMS	EQU	\$7103	JA03
7112	PUTCHR	EQU	\$7112	JA112
7091	LSTTRM	EQU	\$7091	JA11
7082	EOL	EQU	\$7082	JA02
711E	PCRLF	EQU	\$711E	JA024
7127	GETFIL	EQU	\$7127	JA03D
712D	SETEXT	EQU	\$712D	JA033
713C	RPTERR	EQU	\$713C	JA03F
7118	PSTRNG	EQU	\$7118	JA1E

* PROGRAM STARTS HERE

7600		ORG	\$7600	JA100
7600 20 01	CON	BRA	CON1	
7602 01	VN	FCB	1	VERSION NUMBER
7603 B6 78 91	CON1	LDA A	LSTTRM	CHECK TERM CHAR
7606 81 0D		CMP A	#\$D	IS IT CR?
7608 27 05		BNE	CON2	
760A B1 78 82		CMP A	EOL	IS IT EOL?
760D 26 03		BNE	CON3	
760F 7E 71 03	CON2	JMP	WARMS	
7612 CE 77 40	CON3	LDX	#FCB	POINT TO FCB
7615 BD 71 27		JSR	GETFIL	GET FILE NAME
7618 25 54		BOS	ERPORE	ERROR?
761A 86 01		LDA A	#1	SET OPEN
761C A7 00		STA A	B,X	
761E BD 71 2D		JSR	SET TEXT	SET EXTENSION (TXT)
7621 CE 77 40		LDX	#FCB	POINT TO FCB
7624 BD 78 06		JSR	FNS	CALL TO OPEN
7627 26 2C		BNE	CON9	
7629 CC 77 40	CON4	LDX	#FCB	POINT TO FCB
762C BD 78 06		JSR	FMS	READ CHARACTER
762F 20 15		BNE	C1-N6	ERROR?

CONCATENATE FILES UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 2

7631 81 0D		CMP A #\$D	IS IT CR?
7633 26 05		BNE CON5	
7635 BD 71 1E		JSR PCRLF	DO CR & LF
7638 20 EF		BRA CON4	REPEAT
763A 81 0A	CONS	CMP R #\$A	IS IT LINEFEED?
763C 27 EB		BEQ CON4	SKIP IF SO
763E 40		TST R	IS IT NULL?
763F 27 E8		BEQ CON4	SKIP IF SO
7641 BD 71 12		JSR PUTCHR	OUTPUT CHARACTER
7644 20 E3		BRA CON4	REPEAT
7646 A6 01	CON6	LDA R 1,X	GET ERROR NUM
7648 81 08		CMP A #8	IS IT EOF?
764A 26 09		BNE CON9	REPORT ERROR
764C 86 04		LDA R #4	CLOSE FILE
764E A7 00		STR R 0,X	
7650 BD 78 06		JSR FMS	
7653 27 AE		BEQ CON1	REPEAT IF NO ERROR
7655 A6 01	CON9	LDA R 1,X	GET ERROR TYPE
7657 81 03		CMP A #3	FILE EXISTS?
7659 27 09		BEQ ERROR2	
765B 81 04		CMP A #4	NO FILE?
765D 27 0A		BEQ ERROR4	
765F BD 71 3C		JSR RPTERR	REPORT ERROR
7662 20 10		BRA ERROR8	
7664 CE 76 99	ERROR2	LDX #FEST	POINT TO STRING
7667 20 08		BRA ERROR7	
7669 CE 76 7A	ERROR4	LDX #NSST	POINT TO STRING
766C 20 03		BRA ERROR7	
766E CE 76 87	ERROR6	LDX #ILST	POINT TO STRING
7671 BD 71 18	ERROR7	JSR PSTRNG	PRINT STRING
7674 BD 78 03	ERROR8	JSR FMSCLS	CLOSE ALL
7677 7E 71 03		JMP WARMS	

* STRINGS

767A 4E	NSST	FCC	'NO SUCH FILE'
7686 04		FCB	4
7687 49	ILST	FCC	'ILLEGAL FILE NAME'
7698 04		FCB	4
7699 46	FEST	FCC	'FILE EXISTS'
76A4 04		FCB	4
	END	CON	

NO ERROR(S) DETECTED

SYMBOL TABLE:

CON 7600	CON1 7603	CON2 760F	CON3 7612	CON4 7629
CON5 763A	CON6 7640	CON9 7655	EOL 7652	ERROR4 7664
ERROR4 7669	ERROR6 766E	ERROR7 7671	ERROR8 7674	FCB 7740
FEST 7699	FMS 7806	FMSCL 7803	GETFIL 7127	ILST 7687
LSTTRM 7091	NSST 767A	PCRLF 711E	PSTRNG 7118	PUTCHR 7112

CONTIN

The CONTIN command is intended for use in repeating or complex EXEC command files. It prompts the terminal for a YES or NO response for continuing the files execution.

DESCRIPTION

The general syntax of the CONTIN command is:

CONTIN

Executing CONTIN will cause the message 'CONTINUE (Y-N)? ' to be displayed on the terminal. A 'Y' response will cause the EXEC program to execute the next command in the command file. A 'N' response will cause FLEX to regain control and the EXEC program will be halted. This utility is useful for incorporating into EXEC command files which repeat themselves (by calling itself as the last line of the command file). The CONTIN command provides a mechanism for escape from this ever repeating type of command file.

```

*
* CONTINUE UTILITY
*
*
* COPYRIGHT (C) 1978 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*

```

* EQUATES

70A5	CMFLG	EQU	\$70A5	<i>\$AC20</i>
7118	PSTRNG	EQU	\$7118	<i>\$AD1E</i>
710F	GETCHR	EQU	\$710F	<i>\$AD15</i>
7103	WARMS	EQU	\$7103	<i>\$AD03</i>
7600		ORG	\$7600	<i>\$A100</i>

* PROGRAM STARTS HERE

7600 20 01	CON	BRA	CON1	
7602 01	VN	FCB	1	
7603 CE 76 1B	CON1	LDX	#STR	POINT TO STRING
7606 BD 71 19		JSR	PSTRNG	PRINT IT
7609 BD 71 0F		JSR	GETCHR	GET RESPONSE
760C 84 5F		AND A	#\$5F	MAKE UPPER CASE
760E 81 59		CMP A	#`Y	IS IT Y ?
7610 26 03		BNE	CON4	
7612 7E 71 03		JMP	WARMS	CONTINUE
7615 7F 70 A5	CON4	CLR	CMFLG	CLEAR COMMAND MODE
7618 7E 71 03		JMP	WARMS	RETURN TO FLEX
761B 43	STR	FCC	'CONTINUE? '	
7625 04		FCB	4	
		END	CON	

NO ERROR(S) DETECTED

SYMBOL TABLE:

CMFLG	70A5	CON	7600	CON1	7603	CON4	7615	GETCHR	710F
PSTRNG	7118	STR	761B	VN	7602	WARMS	7103		

INTEG

The INTEG command is used to completely test the free space (unused sectors) on a diskette. This routine will guarantee the integrity of the available disk space.

DESCRIPTION

The general syntax of the INTEG command is:

```
INTEG[,<drive number>]
```

where the drive number specifies which disk is to be tested and defaults to the working drive. This program will check that the free space contains the correct number of sectors, that it starts at the correct disk address, and that it terminates at the correct disk address. If any discrepancies are found the appropriate error message will be displayed, otherwise, the message 'FREE SPACE ALL OK!' will be output. An example follows:

```
+---+INTEG,0
```

This would test the free space on the disk in drive 0. It should be noted that INTEG may require a moderate amount of time to run. Any diskette not passing the INTEG test should not be used for creating new files. This command does not test any of the disk space being used by files on the disk.

TEST FREE SPACE INTEGRITY

TSC MNEMONIC ASSEMBLER

PAGE 1

```

*
* INTEG UTILITY
*
*
* COPYRIGHT (C) 1978 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*

```

0100		ORG	\$0100	\$/A100
7806	FMS	EQU	\$7806	\$/B4C6
7740	FCB	EQU	\$7740	\$/A84C
7103	WARM\$	EQU	\$7103	\$/A003
713F	GETHEX	EQU	\$713F	\$/AN42
713C	RPTERR	EQU	\$713C	\$/AN3F
7118	PSTRNG	EQU	\$7118	\$/AN1E
709C	WASN	EQU	\$709C	\$/AC0C

* PROGRAM STARTS HERE

0100 20 09	INTEG	BRA	INTEG1
------------	-------	-----	--------

* TEMPORARY STORAGE

0102 01	WN	FCB	1
0103 00 00	TEMP	FDB	0
0105 00 00	FA	FDB	0
0107 00 00	LA	FDB	0
0109 00 00	CNT	FDB	0

010B BD 71 3F	INTEG1	JSR	GETHEX	GET DRIVE NUMBER
010E 24 06		BCC	INT2	ERROR?
0110 CE 02 40		LDX	#ILST	POINT TO STRING
0113 7E 01 B0		JMP	INT52	
0116 FF 01 03	INT2	STX	TEMP	SAVE DRIVE
0119 CE 77 40		LDX	#FCB	POINT TO FCB
011C 5D		TST B		DEFAULT DRIVE?
011D 26 05		BNE	INT22	
011F B6 70 8C		LDA A	WASN	GET ASN DRIVE
0122 20 03		BRA	INT23	
0124 B6 01 04	INT22	LDA A	TEMP+1	GET SPEC DRIVE
0127 A7 03	INT23	STA A	3,X	SAVE DRIVE NO.
0129 86 10		LDA A	#16	OPEN SIR
012B A7 00		STA A	0,X	
012D BD 78 06		JSR	FMS	CALL FMS
0130 26 07		BNE	INT25	ERROR?
0132 86 07		LDA A	#7	GET INFO REC
0134 A7 00		STA A	0,X	

TEST FREE SPACE INTEGRITY

TSC MNEMONIC ASSEMBLER

PAGE 2

0136 BD 78 06		JSR	FMS	CALL FMS
0139 26 78	INT25	BNE	INT6	ERROR?
013B A6 55		LDA A	85, X	GET START ADDRESS
013D B7 01 05		STA A	FA	AND SAVE IT
0140 A6 56		LDA A	86, X	— 93, X(5D)
0142 B7 01 06		STA A	FA+1	— 94, X(5E)
0145 A6 57		LDA A	87, X	GET LAST ADDRESS
0147 B7 01 07		STA A	LA	AND SAVE IT
014A A6 58		LDA A	88, X	— 95, X(5F)
014C B7 01 08		STA A	LA+1	— 96, X(60)
014F A6 59		LDA A	89, X	GET NUMBER SECTORS
0151 B7 01 09		STA A	CNT	AND SAVE IT
0154 A6 5A		LDA A	90, X	— 97, X(61)
0156 B7 01 0A		STA A	CNT+1	— 98, X(62)
0159 6F 00		CLR	0, X	CLEAR FCODE
015B 86 01		LDA A	#1	FAKE STATUS TO READ
015D A7 02		STA A	2, X	
015F B6 01 05		LDA A	FA	GET START ADDRESS
0162 A7 40		STA A	64, X	PUT IN FCB
0164 A7 1E		STA A	30, X	
0166 B6 01 06		LDA A	FA+1	
0169 A7 41		STA A	65, X	
016B A7 1F		STA A	31, X	
016D 6F 22	INT3	CLR	34, X	CLEAR FLAG
016F 6A 22		DEC	34, X	MAKE NEGATIVE → 2X NOP.
0171 BD 78 06		JSR	FMS	CALL FMS
0174 26 19		BNE	INT5	ERRORS?
0176 7D 01 09		TST	CNT	COUNT = 0?
0179 26 05		BNE	INT4	
017B 7D 01 0A		TST	CNT+1	
017E 27 3B		BEQ	INT7	
0180 B6 01 0A	INT4	LDA A	CNT+1	GET COUNT
0183 80 01		SUB A	#1	DEC BY 1
0185 B7 01 0A		STA A	CNT+1	SAVE
0188 24 E3		BCC	INT3	
018A 7A 01 09		DEC	CNT	DO LSB
018D 20 DE		BRA	INT3	
018F A6 01	INT5	LDA A	1, X	GET ERROR NUMBER
0191 81 08		CMP A	#8	IS IT EOF?
0193 26 21		DNE	INT6	
0195 7D 01 0A		TST	CNT+1	CHECK FOR 0 COUNT
0198 26 26		BNE	INT9	ERROR?
019A 7D 01 09		TST	CNT	GET LAST SECTOR
019D 26 21		BNE	INT8	IS IT RIGHT?
019F A6 1E		LDA A	30, X	
01A1 E6 1F		LDA B	31, X	
01A3 B1 01 07		CMP A	LA	
01A6 26 1D		BNE	INT9	
01A8 F1 01 08		CMP B	LH+1	
01AB 26 18		BNE	INT9	
01AD CE 01 CR		LDA	#OKST	POINT TO STRING
01BD BD 71 18	INT52	JSR	PUTRNG	OUTPUT STRING
01B3 7E 71 03	INT55	JMP	WHRMS	RETURN TO FLEX
01B6 BD 71 3C	INT6	JSR	RPTERR	REPORT ERROR

TEST FREE SPACE INTEGRITY

TSC MNEMONIC ASSEMBLER

PAGE 3

01B9	20 F8		BRA	INT55	
01BB	CE 01 DD	INT?	LDX	#CNTZRO	POINT TO STRING
01BE	20 F0		BRA	INT52	
01C0	CE 02 02	INT8	LDX	#CNTNTZ	POINT TO STRING
01C3	20 EB		BRA	INT52	
01C5	CE 02 20	INT9	LDX	#LANOT	POINT TO STRING
01C8	20 E6		BRA	INT52	

* STRINGS

01CA	46	OKST	FCC	'FREE SPACE ALL OK!'	
01DC	04		FCB	4	
01DD	53	CNTZRO	FCC	'SECTOR COUNT ZERO - NOT END OF SPACE'	
0201	04		FCB	4	
0202	45	CNTNTZ	FCC	'ENCOUNTERED LAST SECTOR EARLY'	
021F	04		FCB	4	
0220	45	LANOT	FCC	'ERROR IN LAST AVAILABLE POINTER'	
023F	04		FCB	4	
0240	49	ILST	FCC	'ILLEGAL DRIVE NUMBER'	
0254	04		FCB	4	
		END	INTEG		

NO ERROR(S) DETECTED

SYMBOL TABLE:

CNT	0109	CNTNTZ	0202	CNTZRO	01DD	FR	0105	FCB	7740
FMS	7806	GETHEX	713F	ILST	0248	INT2	0116	INT22	0124
INT23	0127	INT25	0139	INT3	0160	INT4	0180	INT5	018F
INT52	01B0	INT55	01B3	INT6	01B6	INT7	01BB	INT8	01C0
INT9	01C5	INTEG	01B0	INTEG1	01B8	LA	01B7	LANOT	0220
OKST	01CA	PSTRNG	7118	PPTERR	713C	TEMP	0103	VN	0102
WARM5	7103	WRSN	708C						

RECOVER

The RECOVER command allows the recovery of a file from a disk whose directory has been damaged. It is necessary to know the starting disk address of the file to be recovered.

DESCRIPTION

The general syntax of the RECOVER command is:

```
RECOVER,<disk address>,<file spec>
```

where the file spec defaults to a TXT extension and is forced to drive 0. The bad disk must be in drive 1 and the good disk in drive 0. The disk address is specified as a 4 digit hex number, such that the address track 3, sector 10, would be specified as 030A. This routine starts reading the file on drive 1 at the specified address and copies it into a file on drive 0 giving it the designated name. An example follows:

```
+++RECOVER,1103,INVEN
```

This would start reading data from drive 1 at track 17 (hex 11), sector 3, and copy it into a file named INVEN.TXT on drive 0. This process continues until an end of file is encountered.

```

*
* RECOVER FILE
*
*
* COPYRIGHT (C) 1978 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*

```

* EQUATES

7806	FMS	EQU	\$7806	7A4C6
7803	FMSCLS	EQU	\$7803	7A4C3
7103	WARM5	EQU	\$7103	7ADc3
713F	GETHEX	EQU	\$713F	7AD42
710F	GETCHR	EQU	\$710F	7AD15
713C	RPTERR	EQU	\$713C	7AD3F
7127	GETFIL	EQU	\$7127	7AD2A
7118	PSTRNG	EQU	\$7118	7AD1E
7740	FCB	EQU	\$7740	7AB40
712D	SETEXT	EQU	\$712D	7AN33
0100	ORG		\$0100	7A1C0

* PROGRAM STARTS HERE

0100 20 03	REC	BRA	REC2	
------------	-----	-----	------	--

* TEMPORARY STORAGE

0102 01	VN	FCB	1	
0103 00	TRACK	FCB	0	
0104 00	SECTOR	FCB	0	
0105 BD 71 3F	REC2	JSR	GETHEX	GET DISK ADDRESS
0108 25 78		BCS	REC7	ERROR?
010A FF 01 03		STX	TRACK	SAVE ADDRESS
010D CE 01 8D		LDX	#STR	POINT TO STRING
0110 BD 71 18		JSR	PSTRNG	PRINT IT
0113 BD 71 0F		JSR	GETCHR	GET RESPONSE
0116 84 5F		AND A	#\$5F	MAKE UPPER CASE
0118 81 59		CMP A	#'Y	IS IT YES?
011A 26 66		BNE	REC7	
011C CE 77 40		LDX	#FCB	POINT TO FCB
011F BD 71 27		JSR	GETFIL	GET FILE NAME
0122 25 55		BCS	ERROR6	ERROR?
0124 86 01		LDA A	#1	SET EXTENSION
0126 BD 71 2D		JSR	SETEXT	
0129 CE 77 40		LDX	#FCB	POINT TO FCB
012C 6F 03		CLR	Z,X	CLEAR DRIVE NUMBER

RECOVER FILE UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 2

012E 86 02		LDA A #2	SET FOR WRITE
0130 A7 00		STA A 0,X	
0132 BD 78 06		JSR FMS	OPEN FOR WRITE
0135 26 29		BNE REC5	ERROR?
0137 CE 01 CF		LDX #FCB2	POINT TO 2ND FCB
013A 6F 00		CLR 0,X	CLEAR FCODE
013C 6F 01		CLR 1,X	
013E 86 01		LDA A #1	SET STATUS TO READ
0140 A7 02		STA A 2,X	
0142 R7 03		STA A 3,X	SET DRIVE #1
0144 B6 01 03		LDA A TRACK	GET START ADDRESS
0147 A7 40		STA A 64,X	
0149 B6 01 04		LDA A SECTOR	
014C A7 41		STA A 65,X	
014E 6F 22		CLR 34,X	SET FOR BINARY READ
0150 CE 01 CF	REC4	LDX #FCB2	POINT TO FCB
0153 BD 78 06		JSR FMS	READ CHARACTER
0156 26 2D		BNE REC9	ERROR?
0158 CE 77 40		LDX #FCB	POINT TO OUTPUT
015B BD 78 06		JSR FMS	WRITE CHARACTER
015E 27 F0		BEQ REC4	ERROR?
0160 R6 01	REC5	LDA A 1,X	GET ERROR TYPE
0162 81 03		CMP A #3	FILE EXISTS?
0164 27 09		BEQ ERROR2	
0166 81 04		CMP A #4	NO FILE?
0168 27 0A		BEQ ERROR4	
016A BD 71 3C		JSR RPTERR	REPORT ERROR
016D 20 10		BRA REC6	
016F CE 01 C3	ERROR2	LDX #FEST	POINT TO STRING
0172 20 08		BRA ERROR7	
0174 CE 01 A4	ERROR4	LDX #NSST	POINT TO STRING
0177 20 03		BRA ERROR7	
0179 CE 01 B1	ERROR6	LDX #ILST	POINT TO STRING
017C BD 71 18	ERROR7	JSR PSTRNG	PRINT STRING.
017F BD 78 03	REC6	JSR FMSCLS	CLOSE ALL FILES
0182 7E 71 03	REC7	JMP WAPMS	RETURN TO FLEX
0185 R6 01	REC8	LDA A 1,X	GET ERROR NUMBER
0187 81 08		CMP A #8	IS IT EOF?
0189 27 F4		BEQ REC6	
018B 20 D3		BRA REC5	

* STRINGS

0180 42	STR	FCC	'BAD DISK IN DRIVE #1? '
01A3 04		FCB	4
01A4 4E	NSST	FCC	'NO SUCH FILE'
01B0 04		FCB	4
01B1 49	ILST	FCC	'ILLEGAL FILE NAME'
01C2 04		FCB	4
01C3 46	FEST	FCC	'FILE EXISTS'
01CE 04		FCB	4

01CF FCB2 RMB 192

320

MEMTEST

The MEMTEST command executes a memory pattern test. This test will detect 99% of all memory problems if allowed to run a sufficient amount of time.

DESCRIPTION

The general syntax of the MEMTEST command is:

```
MEMTEST,<hex start address>,<hex end address>
```

The start address is a hex number stating where in memory the test program should start testing, and the end address is the last location to be tested. The test fills memory with random numbers, goes back and checks the numbers are correct, and then repeats the process. This test should be allowed to run approximately 1 hour for each 4K block of memory being tested. Remember not to specify a memory range which will overlap the test program itself. Each successful pass through the test will be shown by the printing of a '!' on the terminal. An example follows:

```
+++MEMTEST,0,3FFF
```

This would test memory from location 0 through location 3FFF. The system RESET must be used to exit the test program. Reboot the system to use FLEX.

*
 * MEMORY PATTERN... A DIAGNOSTIC PROGRAM
 *
 *
 *
 * COPYRIGHT (C) 1978 BY
 *
 * TECHNICAL SYSTEMS CONSULTANTS, INC.
 * P. O. BOX 2574
 * WEST LAFAYETTE, INDIANA 47906
 *
 *

7600		ORG	\$7600	<i>f A1C 0</i>
7600 20 0F	MEM	BRA	START	
7602 01	YN	FCB	1	VERSION NUMBER
7603 00 00	BEGIN	FDB	0	
7605 00 00	END	FDB	0	
7607 00 00	ADDRES	FDB	0	
7609 00 00	TEMP1	FDB	0	
760B 00 00	TEMP2	FDB	0	
760D 00	TEMP	FCB	0	
760E 00	COLCNT	FCB	0	
760F R1 7C	RND	FDB	\$A17C	<i>f A0 50</i>

* EXTERNAL REFERENCES

7112	OUTCH	EQU	\$7112	<i>f A113</i>
710F	INCH	EQU	\$710F	
7103	WARM3	EQU	\$7103	
711E	PCRLF	EQU	\$711E	
7118	PSTRNG	EQU	\$7118	
713F	GETHEX	EQU	\$713F	
7611 BD 71 3F	START	JSR	GETHEX	GET START ADDRESS
7614 25 56		BCS	EXIT	ERROR?
7616 FF 76 03		STX	BEGIN	SAVE ADDRESS
7619 BD 71 3F		JSR	GETHEX	GET LAST ADDRESS
761C 25 4E		BCS	EXIT	ERROR?
761E FF 76 05		STX	END	SAVE ADDRESS
7621 BD 71 1E	ST00	JSR	PCRLF	OUTPUT A CR & LF
7624 7F 76 0E		CLR	COLCNT	CLEAR COLUMN COUNTER
7627 FE 76 0F	START0	LDX	RND	GET RANDOM SEED
762A FF 76 09		STX	TEMP1	SAVE IT
762D FE 76 11		LDX	RNU+2	
7630 FF 76 0B		STX	TEMP2	
7633 FE 76 03	START1	LDX	BEGIN	SET POINTER TO BEGIN
7636 BD 76 B1	STORE	JSR	RANDOM	GET RANDOM NUMBER
7639 A7 00		STA R	0,1	WRITE IN MEMORY

MEMORY PATTERN TEST

TSC MNEMONIC ASSEMBLER

PAGE 2

763B BC 76 05		CPX	END	ARE WE DONE ?
763E 27 03		BEQ	CONT	
7640 08		INX		BUMP THE POINTER
7641 20 F3		BRA	STORE	REPEAT OPERATION
7643 FE 76 09	CONT	LDX	TEMP1	GET ORIGINAL SEED
7646 FF 76 0F		STX	RND	RESTORE
7649 FE 76 03		LDX	BEGIN	RESET POINTER
764C BD 76 B1	COMP	JSR	RANDOM	GET RANDOM AGAIN
764F A1 00		CMP A	0,X	COMPARE MEM WITH \$55
7651 26 1C		BNE	ERROR	ERROR ??
7653 BC 76 05		CPX	END	FINISHED YET ?
7656 27 03		BEQ	CONT2	
7658 08		INX		BUMP THE POINTER
7659 20 F1		BRA	COMP	REPEAT COMPARE
765B 86 21	CONT2	LDA A	#1!	OUTPUT ! FOR CK PASS
765D BD 71 12		JSR	OUTCH	
7660 B6 76 0E		LDA A	COLCNT	CHECK COUNT
7663 81 40		CMP A	#64	
7665 27 BR		BEQ	ST00	
7667 7C 76 0E		INC	COLCNT	BUMP THE COUNTER
766A 20 BB		BRA	START0	GO REPEAT
766C 7E 71 03	EXIT	JMP	WARMS	RETURN TO FLEX

766F 36	ERROR	PSH A		SAVE A
7670 FF 76 07		STX	ADDRES	SAVE X
7673 BD 71 1E		JSR	PCRLF	OUTPUT A C. R. & L. F.
7676 CE 76 07		LDX	#ADDRES	GET ADDRESS
7679 8D 13		BSR	OUT4HX	OUTPUT ADDRESS
767B 32		PUL A		RESTORE A
767C BD 76 99		JSR	OUT25	OUTPUT WRITTEN WORD
767F BD 76 92		JSR	SPACE	OUTPUT A SPACE
7682 FE 76 07		LDX	ADDRES	POINT TO BAD LOC.
7685 BD 76 90		JSR	OUT2HX	OUTPUT BAD DATA
7688 BD 71 1E		JSR	PCRLF	
768B 7E 76 21		JMP	ST00	GO REPEAT

768E 8D 07	OUT4HX	BSR	OUT2	
7690 8D 05	OUT2HX	BSR	OUT2	
7692 86 20	SPACE	LDA A	#\$20	LOAD UP SPACE
7694 7E 71 12		JMP	OUTCH	OUTPUT IT

7697 A6 00	OUT2	LDA A	0,X	GET CHAR.
7699 36	OUT25	PSH A		SAVE IT
769A 8D 04		BSR	HEXL	
769C 32		PUL A		RESTORE A
769D 08		INX		BUMP THE POINTER
769E 20 04		BRA	HEXR	OUTPUT RIGHT SIDE

76A0 44	HEXL	LSR A		SHIFT CHAR. LEFT
---------	------	-------	--	------------------

MEMORY PATTERN TEST

TSC MNEMONIC ASSEMBLER

PAGE 3

76A1	44		LSR A	
76A2	44		LSR A	
76A3	44		LSR A	
76A4	84 0F	HEXR	AND A #\$F	MASK OFF CHAR.
76A6	8B 30		ADD A #\$30	
76A8	81 39		CMP A #'9	COMPARE IT WITH 9
76AA	23 02		BLS HEXR2	
76AC	8B 07		ADD A #7	ADJUST IT
76AE	7E 71 12	HEXR2	JMP OUTCH	OUTPUT IT
76B1	37	RANDOM	PSH B	SAVE B
76B2	C6 08		LDR B #8	SET COUNT
76B4	B6 76 10	RPT	LDA A RND+1	GET NUMBER
76B7	48		ASL A	SHIFT LEFT
76B8	B8 76 10		EOR A RND+1	
76BB	48		ASL A	
76BC	48		ASL R	
76BD	79 76 0F		ROL RND	
76C0	79 76 10		ROL RND+1	
76C3	5A		DEC B	DEC THE COUNT
76C4	26 EE		BNE RPT	
76C6	33		PUL B	RESTORE B
76C7	B6 76 0F		LDA A RND	GET RANDOM
76CA	39		RTS	
			END	MEM

NO ERROR(S) DETECTED

SYMBOL TABLE:

MEMDUMP

The MEMDUMP command allows the dumping or displaying of a selected portion of memory at the terminal in both hex and ASCII. It is very useful as a diagnostic aid.

DESCRIPTION

The general syntax of the MEMDUMP command is:

```
MEMDUMP[,<start address>]
```

where the start address is the hex value of the address at which dumping should start. If the address is left off of the command line, dumping will begin at address 0000. The display consists of 16 lines of data. Each line starts with the address in memory from which the data is taken. Following the address is 16 bytes of data displayed as 2-digit hex numbers. Finally, 16 ASCII characters are printed which represent the data bytes just printed. All control characters are printed as underscores (_). After each block of 256 bytes (16 lines of data), the program stops and expects an input character. Typing an 'F' will cause the display to move Forward, printing the next sequential 256 bytes. Typing a 'B' will move Backwards, printing the previous 256 byte block. A carriage return will return control to FLEX. All other characters are ignored. An example follows:

```
+++MEMDUMP,A00
```

This would cause memory to be dumped starting at location hex 0A00. The hex and ASCII representations are displayed.

```

*
* MEMORY DUMP UTILITY
*
*
* COPYRIGHT (C) 1978 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*

```

7600		ORG	\$7600	
7139	OUTHEX	EQU	\$7139	
713F	GETHEX	EQU	\$713F	
710F	GETCHR	EQU	\$710F	
7112	PUTCHR	EQU	\$7112	
7118	PSTRNG	EQU	\$7118	
711E	PCRLF	EQU	\$711E	
7103	WARM8	EQU	\$7103	
7600 20 06	MDUMP	BRA	MDUMP1	
7602 01	VN	FCB	1	
7603 00 00	ADDRES	FDB	0	
7605 00	PASS	FCB	0	
7606 00 00	TEMP	FDB	0	
7608 BD 71 3F	MDUMP1	JSR	GETHEX	GET ADDRESS
760B 25 62		BCS	MDUMP6	ERROR?
760D FF 76 03		STX	ADDRES	SAVE ADDRESS
7610 BD 71 1E	LOOP	JSR	PCRLF	OUTPUT CR & LF
7613 FE 76 03		LDX	ADDRES	GET ADDRESS
7616 86 10		LDA A	#16	SET LINE COUNT
7618 B7 76 05		STA A	PASS	
761B C6 10	MDUMP2	LDA B	#16	SET COLUMN COUNT
761D FF 76 06		STX	TEMP	SAVE IT
7620 CE 76 06		LDX	#TEMP	POINT TO ADDR
7623 BD 76 72		JSR	OUTADR	PRINT ADDRESS
7626 BD 76 79		JSR	OUTS	PRINT SPACE
7629 FE 76 06		LDX	TEMP	RESTORE POINTER
762C 37	MDUM25	PSH B		
762D BD 76 76		JSR	OUTH	OUTPUT DATA BYTE
7630 08		INX		BUMP TO NEXT
7631 33		PUL B		
7632 5A		DEC B		DEC THE COUNT
7633 26 F7		BNE	MDUM25	
7635 FE 76 06		LDX	TEMP	RESTORE POINTER
7638 C6 10		LDA B	#16	RESET COUNT
763A A6 00	MDUM27	LDA A	0,A	GET CHARACTER
763C 84 7F		AND A	##7F	MASK MSB
763E 81 1F		CMP A	#\$1F	IS IT CONTROL?

MEMORY DUMP UTILITY

TSC MNEMONIC ASSEMBLER PAGE

7640 22 02	BHI	MDUM28	
7642 86 5F	LDA A	#'_	IF SO, OUTPUT '_
7644 BD 71 12 MDUM28	JSR	PUTCHR	OUTPUT CHARACTER
7647 08	INX		BUMP TO NEXT
7648 5A	DEC B		DEC THE COUNT
7649 26 EF	BNE	MDUM27	
764B BD 71 1E	JSR	PCRLF	OUTPUT CR & LF
764E 7A 76 05	DEC	PASS	DEC LINE COUNT
7651 26 C8	BNE	MDUMP2	
7653 7C 76 03	INC	ADDRES	BUMP ADDRESS
7656 BD 71 0F MDUMP5	JSR	GETCHR	GET INPUT
7659 81 0D	CMP A	#\$D	IS IT CR?
765B 27 12	BEQ	MDUMP6	
765D 84 5F	AND A	#\$5F	MAKE UPPER CASE
765F 81 46	CMP A	#`F	IS IT FORWARD?
7661 27 AD	BEQ	LOOP	
7663 81 42	CMP A	#`B	IS IT BACK?
7665 26 EF	BNE	MDUMP5	
7667 7A 76 03	DEC	ADDRES	DEC THE ADDRESS
766A 7A 76 03	DEC	ADDRES	TO MOVE BACK
766D 20 A1	BRA	LOOP	
766F 7E 71 03 MDUMP6	JMP	WARMS	RETURN TO FLEX

* OUTPUT HEX DIGIT

7672 BD 71 39 OUTADR	JSR	OUTHEX	OUTPUT DIGIT
7675 08	INX		BUMP TO NEXT
7676 BD 71 39 OUTH	JSR	OUTHEX	OUTPUT IT
7679 86 20 OUTS	LDA A	#\$20	SETUP SPACE
767B 7E 71 12	JMP	PUTCHR	OUTPUT IT
	END	MDUMP	

NO ERROR(S) DETECTED

SYMBOL TABLE:

ADDRES 7603	GETCHR 710F	GETHEX 713F	LOOP 7610	MDUM25 7620
MDUM27 763A	MDUM28 7644	MDUMP 7600	MDUMP1 7608	MDUMP2 7613
MDUMP5 7656	MDUMP6 766F	OUTADR 7672	OUTH 7676	OUTHEX 7139
OUTS 7679	PASS 7605	PCRLF 711E	PSTRNG 7110	PUTCHR 7112
TEMP 7606	VN 7602	WARMS 7103		

MEMOVE

The MEMOVE command will move any block of memory to any other specified memory location.

DESCRIPTION

The general syntax of the MEMOVE command is:

```
MEMOVE,<start address>,<end address>,<destination>
```

where all addresses are specified in hex. The start and end addresses specify the bounds of the block of memory to be moved, and the destination address designates the location to where it should be moved. An example follows:

```
+++MEMOVE,400,4FF,1A00
```

This command line would cause the block of memory from location hex 400 through hex 4FF to be moved to location hex 1A00. The data which was from 400 through 4FF will now exist from 1A00 to 1AFF. The original block of data is left unchanged.

```

*
* MEMORY MOVE UTILITY
*
*
* COPYRIGHT (C) 1979 BY
*
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* P. O. BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*
*

```

* EQUATES

713F	GETHEX	EQU	\$713F
7103	WARMS	EQU	\$7103
7118	PSTRNG	EQU	\$7118
7600		ORG	\$7600

* PROGRAM STARTS HERE

7600 20 07	MEM	BRA	MEM1	
7602 01	VN	FCB	1	VERSION NUMBER
7603 00 00	START	FDB	0	
7605 00 00	END	FDB	0	
7607 00 00	DESTIN	FDB	0	
7609 BD 71 3F	MEM1	JSR	GETHEX	GET START ADDR
760C 25 3E		BCS	MEM4	ERROR?
760E FF 76 03		STX	START	
7611 BD 71 3F		JSR	GETHEX	GET END ADDRESS
7614 25 36		BCS	MEM4	ERROR?
7616 FF 76 05		STX	END	SAVE POINTER
7619 BD 71 3F		JSR	GETHEX	GET DESTINATION
761C 25 2E		BCS	MEM4	ERROR?
761E FF 76 07		STX	DESTIN	SAVE IT
7621 B6 76 03		LDA A	START	GET START ADDRESS
7624 F6 76 04		LDA B	START+1	
7627 B1 76 07		CMP A	DESTIN	GREATER?
762A 25 29		BLO	MOVE	
762C 22 05		BHI	MOVE	
762E F1 76 08		CMP B	DESTIN+1	CHECK LSB
7631 23 22		BLS	MOVE	

* MOVE BACK

7633 FE 76 03	MOVE	LDX	START	GET START ADDR
7636 A6 00		LDA A	0, X	GET BYTE
7638 BC 76 05		CPX	END	FINISHED?
763B 27 49		BEO	MEM8	
763D 08		INX		BUMP TO NEXT

MEMORY MOVE UTILITY

TSC MNEMONIC ASSEMBLER

PAGE 2

763E FF 76 03	STX	START	SAVE POINTER
7641 FE 76 07	LDX	DESTIN	GET DESTINATION
7644 A7 00	STA A	0,X	PUT BYTE
7646 08	INX		BUMP TO NEXT
7647 FF 76 07	STX	DESTIN	
764A 20 E7	BRA	MOVB	REPEAT
764C CE 76 8E MEM4	LDX	#ILST	POINT TO STRING
764F BD 71 18	JSR	PSTRNG	PRINT IT
7652 7E 71 03	JMP	WARMs	RETURN TO FLEX

* MOVE FORWARD

7655 B6 76 05	MOVF	LDA A	END	GET END POINTER
7658 F6 76 06		LDA B	END+1	
765B F0 76 04		SUB B	START+1	GET DIFFERENCE
765E B2 76 03		SBC A	START	
7661 FB 76 00		ADD B	DESTIN+1	ADD TO DESTIN
7664 B9 76 07		RDC R	DESTIN	
7667 B7 76 07		STA A	DESTIN	SAVE RESULT
766A F7 76 00		STA B	DESTIN+1	
766D FE 76 05	MOVF2	LDX	END	POINT TO END
7670 A6 00		LDA A	0,X	GET BYTE
7672 BC 76 03		CPX	START	FINISHED?
7675 27 0F		BEO	MEM8	
7677 09		DEX		
7678 FF 76 05		STX	END	
767B FE 76 07		LDX	DESTIN	GET DESTINATION
767E A7 00		STA A	0,X	PUT BYTE
7680 09		DEX		
7681 FF 76 07		STX	DESTIN	SAVE POINTER
7684 20 E7		BRA	MOVF2	REPEAT
7686 FE 76 07	MEM8	LDX	DESTIN	GET DESTINATION
7689 A7 00		STA A	0,X	PUT BYTE
768B 7E 71 03		JMP	WARMs	RETURN TO FLEX

* STRINGS

768E 49	ILST	FCC	'ILLEGAL ADDRESS SPECIFIED'
76A7 04		FCB	4
		END	MEM

NO ERROR(S) DETECTED

SYMBOL TABLE:

DESTIN	7607	END	7605	GETHEX	713F	ILST	768E	MEM	7600
MEM1	7609	MEM4	7610	MEM'S	7606	MOVB	7633	MOVF	7655
MOVF2	766D	PSTRNG	7118	START	7603	VN	7602	WARMs	7103

MEMFILL

The MEMFILL command is used to fill a section of memory with a particular data pattern. This is useful for certain types of program debugging and development.

DESCRIPTION

The general syntax of the MEMFILL command is:

```
MEMFILL,<start address>,<end address>,<fill byte>
```

The addresses should be specified in hex. The fill byte is the hex value (8 bit) which will be used to fill memory between the address bounds designated. If the fill byte is left off the command line, zeroes will be used. Upon completion, control will return to FLEX unless it has been overwritten by the command. An example follows:

```
++MEMFILL,0,1FFF,55
```

This would fill memory from location 0 through location hex 1FFF with hex 55 bytes. Remember not to overwrite the program when specifying the address bounds.

```

*
* MEMORY FILL UTILITY
*
* COPYRIGHT (C) 1978 BY
* TECHNICAL SYSTEMS CONSULTANTS, INC.
* BOX 2574
* WEST LAFAYETTE, INDIANA 47906
*

```

```
* EQUATES
```

103	WARMS	EQU	\$7103	
118	PSTRNG	EQU	\$7118	
13F	GETHEX	EQU	\$713F	
600		ORG	\$7600	
600 20 07	MFILL	BRA	MFILL1	
602 01	VN	FCB	1	VERSION NUMBER
603 00 00	BEGIN	FDB	0	
605 00 00	END	FDB	0	
607 00 00	FILL	FDB	0	
609 BD 71 3F	MFILL1	JSR	GETHEX	GET BEGIN ADDRESS
60C 25 26		BCS	MFILL6	ERROR?
60E FF 76 03		STX	BEGIN	SAVE POINTER
611 BD 71 3F		JSR	GETHEX	GET END ADDRESS
614 25 1E		BCS	MFILL6	ERROR?
616 FF 76 05		STX	END	SAVE POINTER
619 BD 71 3F		JSR	GETHEX	GET FILL CHAR
61C 25 16		BCS	MFILL6	ERROR?
61E FF 76 07		STX	FILL	SAVE IT
621 B6 76 08		LDA A	FILL+1	GET CHARACTER
624 FE 76 03		LDX	BEGIN	POINT TO START
627 A7 00	MFILL4	STA R	0,X	PUT CHARACTER
629 BC 76 05		CPX	END	FINISHED?
62C 27 03		BEQ	MFILL5	
62E 08		INX		BUMP TO NEXT
62F 20 F6		BRA	MFILL4	REPET
631 7E 71 03	MFILLS	JMP	WARMS	RETURN TO FLEX
634 CE 76 3C	MFILL6	LDX	#ST1	POINT TO STRING
637 BD 71 18		JSR	PSTRNG	PRINT IT
63A 20 F5		BRA	MFILL5	

```
* STRINGS
```

763C 49	ST1	FCB	"ILLEGAL HEX NUMBER"
764E 04		FCB	4

END	MFILL
-----	-------