

ATARI
PRELIMINARY
CONFIDENTIAL

1 of 12

DISK UTILITIES

Due to Day

COLLEEN DISK UTILITIES

Disk UTILITIES

```

0001      BYDRES = 0      ***USER RAM RESIDENT VERSION***
E456      BYBRES = 1      ***SYSTEM RAM RESIDENT VERSION***
E459      CID      = $E456
E45C      DRIND    = $E459
E45F      SETVBV   = $E45C
E462      GY9VBV   = $E45F
02E9      XITVBV   = $E462
D20E      MENTOP   = $2E9
0011      IRGEN    = $D20E
0010      BRKKEY   = $11
000A      POKMBK   = $10
0008      D03VEC   = $A
BFFA      WARMST   = B
02E7      CARTBT   = $BFFA
02BE      MEMLO    = $2E7
02E0      BIFLOK   = $2BE
0020      RUNAD    = $2E0
0021      ICHIDZ   = $20
0024      ICDNOZ   = $21
0025      ICBALZ   = $24
002E      ICBALZ   = $25
0021      ICIDNI   = $2E
0021      MAXDEV   = $21
031A      HATABB   = $31A

```

```

E63D      /CID SUBROUTINES
E6B9      COMMENT = $E63D
E69E      GDHAND  = $E6B9
          DEVBRC  = $E69E

```

.TITLE "COLLEEN DISK UTILITIES"

```

*****
FILENAME =DK1: DUV. SRC
*****

```

DISK UTILITY ROUTINE FOR COLLEEN D.B.

```

HILD      .MACRO  &1
&1H      =      &1/256
&1L      =      -256*&1H*&1

```

.ENDM

ZERO PAGE VARIABLES

```

001B      JNPTBL   **=11B
001A      RAMLO    **+=2
          **+=2

```

```

1700      VERDOS   = $1700
0700      FMS      = $700
0700      FMIN17   = FMS+$6B

```

COLLEEN DISK UTILITIES

```

      IF      BYBRES
      *-DOB
1300 A7 25      LDA      #DOSBL
1302 B5 0A      STA      DOBVEC
1304 A9 16      LDA      #DOBOSH
1306 B5 08      STA      DOBVEC+1
1308 20 88 07   JBR      FMINIT
130B A5 08      LDA      WARMST  ON COLDSTART, LOAD AUTO. SYS
130D D0 0D      BNE      INITX
130F A9 1D      LDA      #AFL
1311 BD 54 03   STA      ICBAL+010
1314 A9 13      LDA      #AFH
1316 BD 55 03   STA      ICBAL+010
1319 4C 08 24   LOADV   JMP      LOAD  LOAD VECTOR, ALSO INIT LOAD AND EXIT.
131C 60         INITX   RTS
131D 44 31 3A   AF      .BYTE  "D1: AUTD. SYS", CR
1320 41 55 54
1323 4F 2E 53
1326 B9 53 78

```

```

      HILD      AF
0013          AFH      =      AF/256
001D          AFL      =      -256*AFH+AF

```

```

      .ENDIF
      IF      BYBRES-1
      *-USERDOS
      LOADV   =      #1317
      .ENDIF

```

```

007B          CR       =      #7B
001C          CUP      =      #1C
001D          CDN      =      #1D
001E          CLF      =      #1E
001F          CRT      =      #1F
007C          D.L      =      #7C
007D          CLBCR    =      #7D

```

```

0003          OPEN     =      #03
000C          CLOSE    =      #0C
000A          PUTCHR   =      #0A
0006          GETCHR   =      #06
0004          GETRCD   =      #04
000B          PUTRCD   =      #0B
0020          RENAME   =      #20
0021          DELETE   =      #21
00FE          FORMAT   =      #FE
0023          LOCK     =      #23
0024          UNLOCK   =      #24

```

```

0001          TEXT     =      1

```

```

0300          DCB      =      #300
0301          DUNIT    =      DCB+1
0302          DCONND    =      DCB+2
0303          DSTATB    =      DCB+3
0304          DBUFLO    =      DCB+4
0305          DBUFHI    =      DCB+5
0306          DCB      =      DCB+6

```

COLLEEN DISK UTILITIES

030B	DEBII	=	DCB+0B
0340	IOCB	=	0340
0340	ICHID	=	IOCB+0
0341	ICDND	=	IOCB+1
0342	ICCOM	=	IOCB+2
0343	ICBTA	=	IOCB+3
0344	ICBAL	=	IOCB+4
0345	ICBAI	=	IOCB+5
0346	ICBLL	=	IOCB+6
0347	ICBLH	=	IOCB+7
034A	ICAX1	=	IOCB+10
034B	ICAX2	=	IOCB+11
1327	PAR	==+40	PARAMETER AREA
0013	PARH	=	PAR/256
0027	PARL	=	-256*PARH+PAR
1351	LINE	==+80	TYPEIN LINE BUFFER
0013	LBUFH	=	LINE/256
0051	LBUFL	=	-256*LBUFH+LINE
13A1	DBUF	==+100	DATA BUFFER FOR COPY
1421	DB1	=	DBUF+80
137E	DB3	=	DBUF-3
	HILO		DBUF
0013	DBUFH	=	DBUF/256
00A1	DBUFL	=	-256*DBUFH+DBUF
	HILO		DB1
0014	DB1H	=	DB1/256
0021	DB1L	=	-256*DB1H+DB1
	HILO		DB3
0013	DB3H	=	DB3/256
009E	DB3L	=	-256*DB3H+DB3
0000	DBLL	=	0
0001	DBLH	=	1 DATA BUFFER LENGTH=100
14A1	MENUSZ	==+1	
14A2	PER	==+1	
14A3	UNND	==+1	
14A4	RCNT	==+1	
14A5	BSTAT	==+1	
14A6	BNDP	==+5	
14AB	CSRC	==+1	
14AC	CDES	==+1	
14AD	SAVX	==+1	
14AE	PTR	==+1	
14AF	IPTR	==+1	
14B0	CTR	==+1	
14B1	OPT	==+1	
14B2	T1	==+2	
14B4	BTVEC	==+2	A TEMP OF SOME KIND
14B6	MNA	==+2	MENU ADDRESS
14B8	MNL	==+2	MENU LENGTH
14BA	RDM	==+1	FF TO REDISPLAY MENU
14BB	DTH	=	
	HILO		DTH
0014	DTHH	=	DTH/256
00BB	DTHL	=	-256*DTHH+DTH
14BB	LDST	==+2	
14BD	LPND	==+2	

COLLEEN DISK UTILITIES

14BF 84 09	BAVH	. BYTE	684, 609
0014	HIL0		BAVH
00BF	BAVH	=	BAVH/256
14C1 45 3A 9B	BAVH	=	-256*BAVH+BAVH
0014	EDN	. BYTE	"E: ", CR
00C1	EDH	=	EDN/256
14C4 4B 3A 9B	EDL	=	-256*EDH+EDN
	KDN	. BYTE	"K: ", CR
	HIL0		KDN
0014	KDNH	=	KDN/256
00C4	KDNL	=	-256*KDNH+KDN

14C7 7D	DHENU	. BYTE	CLBCR
14CB 44 49 53		. BYTE	"DISK OPERATING SYSTEM 7/24/79", CR

14CB 4B 20 4F
14CE 50 45 52
14D1 41 54 49
14D4 4E 47 20
14D7 53 59 53
14DA 54 45 4D
14DD 20 20 20
14E0 39 2F 32
14E3 34 2F 37

14E6 39 9B		. BYTE	"COPYRIGHT 1979 ATARI", CR CR
------------	--	--------	-------------------------------

14EB 43 4F 50
14EB 59 52 49
14EE 47 4B 54
14F1 20 31 39
14F4 37 39 20
14F7 41 54 41
14FA 52 49 9B
14FD 9B

14FE 41 2E 20		. BYTE	"A. DISK DIRECTORY I. FORMAT DISK", CR
---------------	--	--------	--

1501 44 49 53
1504 4B 20 44
1507 49 52 45
150A 43 54 4F
150D 52 59 20
1510 49 2E 20
1513 46 4F 52
1516 4D 41 54

1519 20 44 49		. BYTE	"B. RUN CARTRIDGE J. DUPLICATE DISK", CR
---------------	--	--------	--

151C 53 4B 9B
151F 42 2E 20
1522 52 55 4E
1525 20 43 41
1528 52 54 52
152B 49 44 47
152E 45 20 20
1531 4A 2E 20
1534 44 55 50
1537 4C 49 43
153A 41 54 45
153D 20 44 49
1540 53 4B 9B
1543 43 2E 20
1546 43 4F 50
1549 50 50 4A

1543 43 2E 20		. BYTE	"C. COPY FILE K. BINARY SAVE", CR
---------------	--	--------	-----------------------------------

COLLEEN DISK UTILITIES

154F 20 20 20
 1552 20 20 20
 1555 4B 2E 20
 1558 42 49 4E
 155B 41 52 59
 155E 20 53 41

1561 56 45 9B
 1564 44 2E 20
 1567 44 45 4C
 156A 45 54 45
 156D 20 46 49
 1570 4C 45 2B
 1573 53 29 20
 1576 4C 2E 20
 1579 42 49 4E

157C 41 52 59
 157F 20 4C 4F

1582 41 44 9B
 1585 45 2E 20
 1588 52 45 4E
 158B 41 4D 45
 158E 20 46 49
 1591 4C 45 20
 1594 20 20 20

1597 4D 2E 20
 159A 52 55 4E
 159D 20 41 54

15A0 20 41 44
 15A3 44 52 45
 15A6 53 53 9B

15A9 46 2E 20
 15AC 4C 4F 43
 15AF 4B 20 46

15B2 49 4C 45
 15B5 20 20 20
 15B8 20 20 20

15BB 4E 2E 20
 15BE 44 45 46
 15C1 49 4E 45

15C4 20 44 45
 15C7 56 49 43
 15CA 45 9B

15CC 47 2E 20
 15CF 55 4E 4C
 15D2 4F 43 4B

15D5 20 46 49
 15DB 4C 45 20
 15DB 20 20 20

15DE 4F 2E 20
 15E1 44 55 50
 15E4 4C 47 43

15E7 41 54 45
 15EA 20 46 49
 15ED 4C 45 9B

15F0 4B 2E 20
 15F3 57 52 49
 15F6 54 43 20

15F9 44 4F 53
 15FC 20 46 49

. BYTE "D. DELETE FILE(S) L. BINARY LOAD", CR

. BYTE "E. RENAME FILE M. RUN AT ADDRESS", CR

. BYTE "F. LOCK FILE N. DEFINE DEVICE", CR

. BYTE "G. UNLOCK FILE O. DUPLICATE FILE", CR

. BYTE "H. WRITE DOS FILE", CR

COLLEEN DISK UTILITIES

1602 1D 1D 1D . BYTE CDN, CDN, CDN, CDN, CDN

1605 1D 1D

1607

DMEND

==

0140

DULEN

=

HILO

DMEND-DMENU

DULEN

0001

DULENI

=

DULEN/256

0040

DULEN

=

-256*DULENI+DULEN

HILO

DMENU

0014

DMENUH

=

DMENU/256

00C7

DMENU

=

-256*DMENUH+DMENU

1607 2D 17 F5

DUJPT

. WORD

DIRLST, BTCAR, CPYFIL, DELFIL, RENFIL, LKFIL, UFIL

160A 1A 3A 19

160D 8A 17 5A

1610 1A 08 1D

1613 30 1D

1615 39 1C 87

. WORD

WBOOT, FMTDSK, DUPDSK, SAVFIL, LDFIL, BRUN, DEFDEV

1618 1A CC 1D

161B DB 20 B1

161E 1C 2C 1B

1621 76 1B

1623 1F 20

. WORD

DUPFIL

HILO

DUJPT

0016

DUJPTH

=

DUJPT/256

0007

DUJPTL

=

-256*DUJPTH+DUJPT

000F

DUNUM

=

15 NUMBER OF FUNCTIONS

DISK UTILITY ROUTINE STARTS HERE

DISK OPERATING SYS MONITOR

1625 A9 FF

DOSOS

LDA

#FF

HILO

DOSOS

0016

DOSOSH

=

DOSOS/256

0025

DOSOSL

=

-256*DOSOSH+DOSOS

1627 B5 11

STA

BRKKEY

1629 B5 0B

STA

WARNST

162B A5 10

LDA

POKMSK ENABLE BREAK INTERRUPTS

162D 09 80

ORA

#80

162F B5 10

STA

POKMSK

1631 8D 0E D2

STA

IRGEN

1634 20 BC 21

JBR

INITIO CLOSE FILES

1637 A9 03

LDA

#OPEN

1639 8D 72 03

STA

ICCDM+30 K: IOCB

163C A2 30

LDX

#30 K: IOCB

163E A9 04

LDA

#4

1640 7D 4A 03

STA

ICAX1, X OPEN READ

1643 A9 14

LDA

#KDNH

1645 7D 45 03

STA

ICBAH, X

1648 A9 C4

LDA

#KDNL

164A 7D 4A 03

STA

ICBAL, X

COLLEEN DISK UTILITIES

DISK UTILITY MONITOR

```

1650          DBKUTL
1650 A9 0F    DUJ    LDA    #DUNUM
1652 BD A1 14    STA    MENU8Z    SET MENU SIZE.
1655 A9 07    LDA    #DUJPTL
1657 B9 18    STA    JMPTBL
1659 A9 16    LDA    #DUJPTH
165B B5 19    STA    JMPTBL+1    SET UP JUMP TABLE ADDRESS
165D A9 C7    LDA    #DMENUL
165F BD B6 14    STA    MNA    SET UP FOR MENU COPY
1662 A9 14    LDA    #DMENULH
1664 BD B7 14    STA    MNA+1
1667 A9 40    LDA    #DULENL
1669 BD B8 14    STA    MNL
166C A9 01    LDA    #DULENH
166E BD B9 14    STA    MNL+1

```

FALL THRU TO MENU SELECT

MENU SELECT MONITOR -- VECTORS TO ROUTINE SELECTED FROM MENU.

```

1671 A9 00    BNMEN LDA    #0    CLEAR PAUSE AND SHOW MENU FLAG
1673 BD BA 14    STA    RDM
1676 AD B6 14    LDA    MNA
1679 BD 44 03    STA    ICBAL
167C AD B7 14    LDA    MNA+1
167F BD 45 03    STA    ICBALH
1682 AD B8 14    LDA    MNL
1685 BD 48 03    STA    ICBLL
1688 AD B9 14    LDA    MNL+1
168B BD 49 03    STA    ICBLLH
168E 20 56 23    JSR    DBPM80    SHOW MENU

```

SELECT ITEM FROM MENU
FUNCTION COME HERE WHEN THEY ARE DONE.

```

1691 A2 FF    MENUBL LDX    #0FF    RESET STACK AT THIS POINT
1693 7A      TXS
1694 2C BA 14    BIT    RDM    IF WE SHOULD SHOW MENU
1697 30 62    BMI    PSMEN    GO DO IT
1699 A9 EF    LDA    #81TL    SELECT ITEM MESSAGE
169B A2 16    LDX    #81TH
169D 20 61 23    JSR    DBPLIN
16A0 A9 40    LDA    #040    MAKE SURE UPPER CASE
16A2 BD BE 02    STA    SHFLOK
16A5 20 33 22    JSR    CHROET    GO GET KEYBOARD CHAR.
16A8 38      BEC
16A9 E9 41    BNC    #'A    CONVERT ASCII CHAR. TO BINARY & SUBTRACT 1.
16AB 30 28    BMI    RANGE    IF ASCII CHAR NOT A 0, GO READ AGAIN
16AD CD A1 14    CMP    MENU8Z    IS THE # ENTERED > MENU SIZE?
16B0 10 26    BPL    RANGE    IF YES, GO READ AGAIN.
16B2 0A      ASL    A
16B3 AB      TAY
16B4 B1 18    LDA    (JMPTBL),Y
16B6 CB      INY
16B7 B5 1A    STA    RAMLO    GET STRING POINTER

```

COLLEEN DISK UTILITIES

168B 85 1B	STA	RAMLO+1	
168D A0 01	LDY	#1	LOAD STRING POINTER INTO REGISTERS
168F B1 1A	LDA	(RAMLO),Y	FOR DSPLIN
16C1 AA	TAX		
16C2 8B	DEY		
16C3 B1 1A	LDA	(RAMLO),Y	
16C5 20 61 23	JSR	DSPLIN	PRINT MODULES INITIAL STRING
16C8 A5 1A	LDA	RAMLO	INC BY 2 TO POINT PAST STRING POINTER
16CA 19	CLC		
16CB 69 02	ADC	#2	
16CD 85 1A	STA	RAMLO	
16CF A5 1B	LDA	RAMLO+1	
16D1 69 00	ADC	#0	CARRY
16D3 85 1B	STA	RAMLO+1	PUT HI BYTE.
16D5 6C 1A 00	JMP	(RAMLO)	JUMP TO ROUTINE SELECTED BY MENU.
16D8 A9 E2	RANGE LDA	#NSIL	
16DA A2 16	LDX	#NSIH	
16DC 20 61 23	JSR	DSPLIN	NO SUCH ITEM MESSAGE
16DF 4C 91 16	JMP	MENUSL	
16E2 4E 4F 20	NSI . BYTE	"NO SUCH ITEM", CR	
16E5 53 55 43			
16E8 4B 20 49			
16EB 54 45 4D			
16EE 9B			
16EF 53 45 4C	BIT . BYTE	"SELECT ITEM", CR	
16F2 45 43 54			
16F5 20 49 54			
16FB 45 4D 9B			

0016	NSIH	=	NSI
00E2	NSIL	=	NSI/256
		=	-256*NSIH+NSI
0016	BITH	=	BIT
00EF	BITL	=	BIT/256
		=	-256*BITH+BIT
1691	MNSL	=	MENUSL
		=	MNSL
0016	MNSLH	=	MNSL/256
0091	MNSLL	=	-256*MNSLH+MNSL

PAUSE AND REDISPLAY MENU

16FB A9 00	PBMEN LDA	#0
16FD 8D 49 03	STA	ICBLH
1700 A9 15	LDA	#21
1702 8D 4B 03	STA	ICRL
1705 A9 17	LDA	#PMESH
1707 8D 45 03	STA	ICBAH
170A A9 1B	LDA	#PMESL
170C 8D 44 03	STA	ICBAL
170F 20 56 23	JSR	DSPMS0
1712 20 33 22	JSR	CHROET
1715 4C 71 16	JMP	BHMEN
171B 9B 54 59	PMES . BYTE	CR, "TYPE RETURN FOR MENU"
171B 50 45 20		
171E 52 45 54		
1721 55 52 4E		
1724 20 46 4F		
1727 52 20 4D		
172A 45 4E 55		

COLLEEN DISK UTILITIES

```

0017      PHEB1  =      PHEB/256
0018      PHEB1  =      -256*PHEB1+PHEB

```

```

;
;      DIRECTORY LISTING ROUTINE
;
;*****

```

```

172D 9B 17      DIRLST  .WORD  DLMO
172F 20 B4 22      JSR      GETIC1
1732 AE AE 14      LDX      PTR
1735 A9 9B      LDA      #CR
1737 9D 2B 13      STA      PAR-1,X ASSURE GOOD TERM
173A BD 27 13      LDA      PAR-2,X LAST CHAR OF SEARCH SPEC
173D C9 3A      CMP      #' IF COLON, ADD *,*
173F D0 1B      BNE      QLF
1741 A9 2A      LDA      #'*
1743 9D 2B 13      STA      PAR-1,X
1746 9D 2A 13      STA      PAR+1,X
1749 A9 2E      LDA      #'
174B 9D 29 13      STA      PAR,X
174E A9 9B      LDA      #CR
1750 9D 2B 13      STA      PAR+2,X
1753 E8      INX
1754 E8      INX
1755 E8      INX
1756 BE AE 14      STX      PTR
1759 BE AD 14      QLF      STX      SAVX
175C A2 20      LDX      #*20
175E 20 B1 23      JSR      PIOC8
1761 20 9E 22      JSR      GETFIL
1764 20 79 22      JSR      PERX
1767 A9 06      LDA      #6 READ DIR INFO
1769 A2 10      LDX      #*10
176B 9D 4A 03      STA      ICAX1,X
176E A9 03      LDA      #OPEN OPEN
1770 9D 42 03      STA      ICCOM,X
1773 BE AB 14      STX      CBRC COPY SOURCE-DIRECTORY INFO
1776 E0 10      CPX      #*10
1778 D0 01      BNE      #+3
177A 20 C2 23      JSR      CIOCL
177D AD AE 14      LDA      PTR
1780 3B      BEC
1781 ED AD 14      BDC      SAVX
1784 C9 03      CMP      #3 IF ONLY 3 CHARS, IS "D:"CR, USE DEFAULT
1786 F0 03      BEQ      DLST1
1788 4C 9B 19      DLST0  JH*      PDEB GO INTO COPY
178B AE AD 14      DLST1  LDX      SAVX
178E BD 27 13      LDA      PAR,X
1791 C9 44      CMP      #'D
1793 D0 F3      BNE      DLST0
1795 4C A2 19      JH*      PDEB1 GO INTO COPY WITH DEB="E:"
1798 44 49 32      DLMO  .BYTE  "DIRECTORY--SEARCH SPEC,LIST FILE?",CR
179B 45 43 34
179E 4E 32 39
17A1 2D 2D 33
17A4 45 41 32
17A7 43 4B 20
17AA 53 50 45
17AD 43 2C 4C

```

COLLEEN DISK UTILITIES

17B3 20 46 49
17B6 4C 45 3F
17B9 9B

DELETE FILE ROUTINE

17BA FC 18	DELFIL	.WORD	DEMO
17BC 20 84 22		JBR	QETIC1
17BF 20 79 22		JBR	PERX
17C2 AD 29 13		LDA	PAR
17C5 C9 44		CMF	#D
17C7 F0 1A		BEG	DF1
17C9 A9 D3		LDA	#NDFL
17CB A2 17		LDX	#NDFH
17CD 20 61 23		JBR	DBPLIN
17D0 4C 91 16		JMP	MENUSL
17D3 4E 4F 54	NDF	.BYTE	"NOT A DISK FILE",CR
17D6 20 41 20			
17D9 44 49 53			
17DC 4B 20 46			
17DF 49 4C 45			
17E2 9B			

0017		HIL0	NDF
00D3	NDFH	=	NDF/256
	NDFL	=	-256*NDFH+NDF
17E3 A2 10	DF1	LDX	#10
17E5 AD 81 14		LDA	OPT
17E8 C9 4E		CMF	#N
17EA D0 0B		BNE	DW0
17EC A9 21		LDA	#DELETE
17EE 9D 42 03		STA	ICCOM.X
17F1 20 C2 23		JBR	CIOCL
17F4 4C 91 16		JMP	MENUSL
17F7 A9 E6	DW0	LDA	#TYGL
17F9 A2 1B		LDX	#TYQH
17FB 20 61 23		JBR	DBPLIN
17FE A9 00		LDA	#0
1800 BD AF 14		STA	IPTR
1803 A2 20		LDX	#20
1805 A9 21		LDA	#DELETE
1807 7D 42 03		STA	ICCOM.X
180A A9 9E		LDA	#DB3L
180C 9D 44 03		STA	ICBAL.X
180F A9 13		LDA	#DB3H
1811 9D 45 03		STA	ICBAH.X
1814 A9 44		LDA	#D
1816 BD 7E 13		STA	DBUF-3
1819 A9 3A		LDA	#:
181B BD A0 13		STA	DBUF-1
181E AD 2A 13		LDA	PAR+1
1821 C9 3A		CMF	#:
1823 D0 02		BNE	++4
1825 A9 31		LDA	#1
1827 BD 9F 13		STA	DBUF-2
182A A2 10	IMRD	LDX	#10
182C A2 02		LDA	#255

DEVICE NUMBER OR : FROM OP' INPUT

KLUDGE KLUDGE KLUDGE

COLLEEN DISK UTILITIES

```

182E 9D 42 03      STA      ICCOM,X
1831 A9 06          LDA      #6
1833 9D 4A 03      STA      ICAX1,X DIR READ OPEN
1836 A9 29          LDA      #PARL
1838 9D 44 03      STA      ICBAL,X
183B A9 13          LDA      #PARH
183D 9D 45 03      STA      ICBAL,X
1840 20 C2 23      JBR      CIOCL
1843 A9 A1          LDA      #DBUFL
1845 9D 44 03      STA      ICBAL,X
1848 A9 13          LDA      #DBUFH
184A 9D 45 03      STA      ICBAL,X
184D A9 04          LDA      #GETRCD
184F 9D 42 03      STA      ICCOM,X
1852 A9 00          LDA      #0
1854 BD AE 14      STA      PTR      HOW MANY FILES WE HAVE SKIPPED

```

```

      J READ FILENAME FROM DIR. QUERY AND DELETE
1857 A2 10      RDFN      LDX      #10
1859 A9 00      LDA      #0
185B 9D 48 03      STA      ICBLL,X
185E A9 01      LDA      #1
1860 9D 49 03      STA      ICBLL,X
1863 20 C2 23      JBR      CIOCL READ A LINE FROM DIRECTORY
1866 AD A2 13      LDA      DBUF+1 IF FILE LINE. THIS IS BLANK

```

```

1869 C9 20      CMF      #'
186B D0 68      BNE      DELX THIS IS FREE BLOCKS LINE
186D EE AE 14      INC      PTR COUNT THIS FILE
1870 AD AE 14      LDA      PTR HAVE WE SKIPPED ENUF YET
1873 CD AF 14      CMF      JPTR
1876 30 DE      BMI      RDFN BR IF NO

```

```

1878 A2 00      LDX      #0 PUT PTR
187A AD 02      LDY      #2 GET PTR

```

```

      MESSAGE DELETE FILE NAME
187C B9 A1 13      MDN1      LDA      DBUF,Y
187F C9 20      CMF      #' END OF FILENAME

```

```

1881 F0 09      BEQ      MDN2
1883 9D A1 13      STA      DBUF,X
1886 EB          JNX
1887 CB          JNY

```

```

188B E0 08      CPX      #8
188A 30 F0      BMI      MDN1

```

```

      FILENAME IS MOVED. PUT EXT

```

```

188C A9 2E      MDN2      LDA      #'
188E 9D A1 13      STA      DBUF,X
1891 EB          JNX
1892 AD 0A      LDY      #10 WHERE EXT IS
1894 B9 A1 13      MDN3      LDA      DBUF,Y
1897 9D A1 13      STA      DBUF,X

```

```

189A CB          JNY
189B EB          JNX
189C C0 0D      CPY      #13
189E 30 F4      BMI      MDN3
18A0 BE AD 14      BVSX      PUT CR HERE LATER
18A3 A9 3F      LDA      #'? FOR QUERY
18A5 9D A1 13      STA      DBUF,X
18AB EB          JNX

```

COLLEEN DISK UTILITIES

```

18B2 20 41 23      JSR    DBPLIN  GO ASK ABOUT THIS FILE
18B5 20 33 22      JSR    CHROET
18B8 C9 59          CMP     #'Y
18BA D0 9B          BNE     RDFN    GO DO NEXT FILENAME
18BC AD AE 14       LDA     PTR     NUMBER FILES WE HAVE GONE THRU SO FAR
18BF 8D AF 14       BTA     IPTR    IS NEW NUMBER TO SKIP.
18C2 AE AD 14       LDX     SAVX
18C5 A9 9B          LDA     #CR
18C7 9D A1 13       BTA     DBUF,X
18CA A2 20          LDX     #20    DELETE IOCB
18CC 20 C2 23       JSR     CIOCL
18CF 20 DB 18       JSR     CIOB1

```

```

18D2 4C 2A 18      JMW     IDRD    CLOSE AND REDPEN DIR READ FILE
18D5 20 DB 18      JSR     CIOB1  CLOSE DIR READ FILE
18D8 4C 91 16      JMW     MENUCL
18DB A2 10          CIOB1  LDX     #10
18DD A9 0C          LDA     #CLOSE
18DF 9D 42 03       STA     ICCOM,X
18E2 20 C2 23       JSR     CIOCL
18E5 60             RTS

```

```

18E6 54 59 50      TYO     .BYTE  "TYPE ",#22,"Y",#22," " TO DELETE,...",CR
18E9 45 20 22
18EC 57 22 20
18EF 54 4F 20

```

```

18F2 44 45 4C
18F5 45 54 45
18F8 2E 2E 2E
18FB 9B

```

```

001B              TYOH    =      TYO
00E6              TYOL    =      -256*TYOH+TYO
18FC 44 45 4C      DEMO    .BYTE  "DELETE FILE SPEC",CR
18FF 45 54 45
1902 20 46 49
1905 4C 45 20
1908 53 50 45
190B 43 9B

```

```

18E6 54 59 50
18E9 45 20 22
18EC 57 22 20
18EF 54 4F 20

```

```

18F2 44 45 4C
18F5 45 54 45
18F8 2E 2E 2E
18FB 9B

```

```

001B              TYOH    =      TYO
00E6              TYOL    =      -256*TYOH+TYO
18FC 44 45 4C      DEMO    .BYTE  "DELETE FILE SPEC",CR
18FF 45 54 45
1902 20 46 49
1905 4C 45 20
1908 53 50 45
190B 43 9B

```

```

; *****
;
;

```

COPY FILE ROUTINE

```

190D 43 4F 50      CPMO    .BYTE  "COPY---FROM, TD?",CR
1910 57 2D 2D
1913 46 52 4F
1916 4D 2C 20
1919 54 4F 3F
191C 9B

```

```

191D 4F 50 54      DE      .BYTE  "OPTION DOESNT MAKE SENSE",CR
1920 47 4E 4E
1923 20 44 4F
1926 45 53 4E
1929 54 20 4D
192C 41 4B 45
192F 20 53 45
1932 4E 53 45
1935 9B

```

HILO DE

COLLEEN DISK UTILITIES

001D		DEL	=	-256*DEH*DE
1936 0D 19		CPYFIL	.WORD	CPMO
1938 20 84 22		JBR	GETIC1	
1938 AD B1 14		LDA	OPT	
193E 4B		PIA		
193F AE AE 14		LDX	PTR	
1942 CA		DEX		
1943 A9 9B		IDA	#CR	TERM FIRST FILENAME
1945 9D 29 13		STA	PAR,X	
1948 A2 20		LDX	#20	IOCB 3
194A 20 B1 23		JBR	PIOCB	
194D AD AE 14		LDA	PTR	
1950 BD AD 14		BTA	SAVX	
1953 20 9E 22		JBR	GETFIL	GET SECOND FILENAME
1956 20 79 22		JBR	PERX	IF PARAM ERRS, EXIT
1959 6B	PBRC	PLA		
195A AA		TAX		
195B AD 29 13		LDA	PAR	GET 1ST LETR OF PARAM
195E C9 4B		CMF	#'K	K AND E AS SOURCE ARE SPECIAL
1960 F0 4B		BEG	ODMS	K: GETB 'OPTION DOESNT MAKE SENSE' FOR NOW
1962 C9 45		CMF	#'E	
1964 D0 0B		BNE	OPBRC	IF NO THEN OPEN SOURCE FILE
1966 CE 8A 14		DEC	RDM	E: SD REBROW MENU AFTER COPY
1969 A2 00		LDX	#0	
196B BE AB 14		BTX	CBRC	
196E 4C 9B 19		JMP	PDEB	
1971 C9 53	OPBRC	CMF	#'S	
1973 F0 3B		BEG	ODMS	S: AS SOURCE GETS O.D.M.S. FOR NOW
1975 E0 53		CPX	#'S	
1977 D0 09		BNE	OPBRC1	
1979 C9 43		CMF	#'C	
197B D0 30		BNE	ODMS	
197D A9 80		LDA	#9B0	
197F BD 5B 03		STA	ICAX2+#10	
1982 A2 10	OPBRC1	LDX	#10	
1984 A9 03		LDA	#OPEN	
1986 9D 42 03		BTA	ICCOM,X	
1989 A9 04		LDA	#4	OPEN IN
198B 9D 4A 03		STA	ICAX1,X	
198E BE AB 14		BTX	CBRC	
1991 E0 10		CPX	#10	
1993 D0 1F		BNE	+133	
1995 20 C2 23		JBR	CIOCL	OPEN SOURCE FILE HERE
199B AE AD 14	PDEB	LDX	SAVX	
199B BD 29 13		LDA	PAR,X	
199E C9 45		CMF	#'E	CHECK FOR SPECIAL CASE
19A0 D0 15		BNE	OPDEB	IF NOT
19A2 A9 00	PDEB1	LDA	#0	SPECIAL CASE - DONT OPEN, USE EXISTING IOCB
19A4 BD AC 14		BTA	CDEB	
19A7 CE 8A 14		DEC	RDM	SET FLAG TO REBROW MENU
19AA 4C DA 19		JMP	DOCPY	
19AD A9 1D	ODMS	LDA	#DEL	
19AF A2 19		LDX	#DEH	
19B1 20 61 23		JBR	DBPLIN	
19B4 4C 91 16		JMP	MENUBL	
19B7 E0 41	OPDEB	CPX	#'A	APPEND TO DISK FILE
19B9 D0 0B		BNE	OPDEB1	
19BB C9 44		CMF	#'D	
19BD D0 EE		DNH	ODMS	

COLLEEN DISK UTILITIES

```

19C1 D0 02          BNE    OPDEB3
19C3 A9 08          OPDEB1 LDA    #B
19C5 A2 20          OPDEB3 LDX    ##20
19C7 9D 4A 03      STA    ICAX1,X OPEN TYPE OUT
19CA A9 03          LDA    #OPEN
19CC 9D 42 03      STA    ICCOM,X OPEN
19CF 8E AC 14      STX    CDEB
19D2 20 C2 23      JSR    CIOCL
19D5 A9 00          LDA    #0
19D7 9D 4B 03      STA    ICAX2,X
          ; COPY FROM CBRC TO CDEB
19DA A9 07          DDCPY LDA    @GETCHR+TEXT
19DC AE AB 14      QC1    LDX    CBRC
19DF AC AC 14      LDY    CDEB
19E2 9D 42 03      STA    ICCOM,X
19E5 A9 08          LDA    @PUTCHR+TEXT
19E7 99 42 03      STA    ICCOM,Y
19EA A9 A1          LDA    @DBUFL
19EC 9D 44 03      STA    ICBAL,X
19EF 99 44 03      STA    ICBAL,Y
19F2 A9 13          LDA    @DBUFH
19F4 9D 45 03      STA    ICBAL,X
19F7 99 45 03      STA    ICBAL,Y
19FA AE AB 14      CLOOP LDX    CBRC
19FD A9 00          LDA    #000
19FF 9D 4B 03      STA    ICBLL,X
1A02 A9 01          LDA    #001
1A04 9D 49 03      STA    ICBLL,X
1A07 20 36 E4      JSR    READ FROM INPUT
1A0A BC A5 14      BTY    BSTAT
1A0D AE AC 14      LDX    CDEB
1A10 AC AB 14      LDY    CBRC
1A13 B9 4B 03      LDA    ICBLL,Y
1A16 9D 4B 03      STA    ICBLL,X
1A19 B9 49 03      LDA    ICBLL,Y
1A1C 9D 49 03      STA    ICBLL,X
1A1F 19 4B 03      DRA    ICBLL,Y IF SOURCE FILE LENGTH = 0
1A22 F0 03          BEQ    CKRS DONT DO WRITE
1A24 20 C2 23      JSR    CIOCL WRITE, ABORT IF ERROR
1A27 AD A5 14      CKRS LDA    BSTAT GET READ OPERATION STATUS BACK
1A2A 10 CE          BPL    CLOOP IF OK, GO READ SOME MORE
1A2C C9 BB          CMV    ##99 EOF STATUS
1A2E F0 03          BEQ    ++5
1A30 4C CA 23      JMP    CIOER IF NOT, ABORT
1A33 AE AB 14      CLOC LDX    CBRC
1A36 F0 0C          BEQ    DU4 IF E:, DONT CLOSE
1A3B E0 30          CPX    ##30 OR K:
1A3A F0 0B          BEQ    DU4
          ; CLOSE SOURCE FILE
1A3C A9 0C          LDA    #CLOSE
1A3E 9D 42 03      STA    ICCOM,X
1A41 20 36 E4      JSR    CIO
1A44 AE AC 14      DU4 LDX    CDEB
1A47 F0 0B          BEQ    DU3 IF DEB=E:
1A49 A9 0C          LDA    #CLOSE
1A4B 9D 42 03      STA    ICCOM,X
1A4E 20 36 E4      JSR    CIO
1A51 4C 91 16      DU3 JMP    MENU/SL

```

VR-1412

COLLEEN DISK UTILITIES

RENAME FILE ROUTINE

1A54 6C 1A	RENFIL	WORD	RNM0
1A56 20 84 22		JSR	QETIC1
1A59 20 90 22		JSR	QETNAME
1A5C 20 79 22		JSR	PERX
1A5F A9 20		LDA	#RENAME
1A61 A2 10		LDX	#*10
1A63 9D 42 03		BTA	ICCOM, X
1A66 20 C2 23		JSR	CIOCL
1A69 4C 91 16		JMP	MENUSL
1A6C 52 45 4E	RNM0	BYTE	"RENAME, GIVE OLD NAME, NEW", CR
1A6F 41 4D 45			
1A72 2C 20 47			
1A75 49 56 45			
1A7B 20 4F 4C			
1A7B 44 20 4E			
1A7E 41 4D 45			
1A81 2C 20 4E			
1A84 45 57 9B			

FORMAT DISK ROUTINE

1A87 C0 1A	FMTDSK	WORD	WHD
1A89 20 F1 21		JSR	QETLIN
1A8C 20 53 25		JSR	QETDN
1A8F 1B		CLC	
1A90 69 30		ADC	#'0
1A92 8D EF 1A		BTA	DDSK
1A95 8D F2 1A		BTA	CDSK
1A98 20 79 22		JSR	PERX
1A9B A9 D7		LDA	#VFML QUERY TO VERIFY DRIVE NUMBER
1A9D A2 1A		LDX	#VFML
1A9F 20 61 23		JSR	DSPLIN
1AA2 20 33 22		JSR	CHROET
1AA5 C9 59		CMR	#'Y SEE IF OK
1AA7 D0 14		BNE	FHX
1AA9 A9 F1		LDA	#FDPL
1AAB A2 10		LDX	#*10
1AAD 9D 44 03		BTA	ICBAL, X
1AB0 A9 1A		LDA	#FDPH
1AB2 9D 45 03		BTA	ICBAH, X
1AB5 A9 FE		LDA	#FORMAT
1AB7 9D 42 03		BTA	ICCOM, X
1ABA 20 C2 23		JSR	CIOCL CALL CIO TO DO FORMAT
1ABD 4C 91 16	FHX	JMP	MENUSL EXIT.
1AC0 57 48 49	WHD	BYTE	"WHICH DRIVE TO FORMAT?", CR
1AC3 43 48 20			
1AC6 44 52 49			
1AC9 56 45 20			
1ACC 54 4F 20			
1ACF 46 4F 52			
1AD2 4D 41 54			

COLLEEN DISK UTILITIES

1AD7 54 59 50 VFM . BYTE "TYPE ",#22,"Y",#22," TO FORMAT DISK "

1ADA 45 20 22

1ADD 59 22 20

1AEO 54 4F 20

1AE3 46 4F 52

1AE6 4D 41 54

1AE9 20 44 49

1AEC 53 4B 20

1AEF

DDSK

==+1

1AF0 9B

. BYTE

CR

1AF1 44

FM

. BYTE

"D"

1AF2

CDISK

==+1

1AF3 3A 9B

. BYTE

" ", CR

001A

WHDH

=

WHD/256

00C0

WHL

=

-256*WHDH+WHD

001A

VFMH

=

VFM/256

00D7

VFHL

=

-256*VFMH+VFM

001A

FDPH

=

FDP/256

00F1

FDPL

=

-256*FDPH+FDP

START CART ROUTINE

E45F

BYVBL

=

BYBVBV

00E4

BYVBLH

=

BYVBL/256

005F

BYVBL

=

-256*BYVBLH+BYVBL

E462

XTVBL

=

XITVBV

HIL

XTVBL

00E4

XTVBLH

=

XTVBL/256

0062

XTVBL

=

-256*XTVBLH+XTVBL

1AF5 2B 1B

BTCAR

WORD

SCMO

1AF7 AD FC BF

LDA

#BFFC

BEE IF CART

1AFA F0 0A

BEG

1BCAR

GO IF THERE IS

1AFC A9 1E

LDA

#NCAL

1AFE A2 1B

LDX

#NCAH

SAY NO CART

1B00 20 61 23

JBR

DBPLIN

1B03 4C 91 16

JM'

MENUSL

1B06 20 BC 21

1BCAR

JBR

INITIO

RESET VERTICAL BLANK VECTOR8 BEFORE ENTERING CART

1B09 A9 06

LDA

#6

SET VVBLKI

1B0B A2 E4

LDX

#BYVBLH

HI BYTE

1B0D A0 5F

LDY

#BYVBL

1B0F 20 5C E4

JBR

SETVBV

1B12 A9 07

LDA

#7

SET VVBLKD

1B14 A2 E4

LDX

#XTVBLH

1B16 A0 62

LDY

#XTVBL

1B1B 20 5C E4

JBR

SETVBV

1B1B 6C FA BF

JM'

(CARTST)

1B1E 4E 4F 20

NCA

. BYTE

"NO CARTRIDGE", CR

1B21 43 41 52

1B24 54 52 49

1B27 44 47 45

1B2A 9B

HIL NCA

COLLEEN DISK UTILITIES

```

001E      NCAL      -256*NCAH+NCA
1B2B 9B    BCHQ     . BYTE  CR

```

```

1B2C 3E 1B    BRUN     . WORD  BRMO
1B2E 20 F1 21    JBR     GETLIN
1B31 20 8A 24    JBR     GETNO
1B34 20 79 22    JBR     PERX
1B37 85 1A      GTA     RAMLO
1B39 86 1B      STX     RAMLO+1
1B3B 6C 1A 00    JMP     (RAMLO)
1B3E 52 55 4E    BRMO     . BYTE  "RUN FROM WHAT ADDRESS?".CR
1B41 20 46 52
1B44 4F 4D 20
1B47 57 4B 41
1B4A 54 20 41
1B4D 44 44 52
1B50 45 53 53
1B53 3F 9B

```

DEFINE DEVICE

```

1B55 4C 4F 47    DEDE     . BYTE  "LOGICAL DEVICE, PHYSICAL DEVICE?".CR
1B58 49 43 41
1B5B 4C 20 44
1B5E 45 56 49
1B61 43 45 2C
1B64 20 50 4B
1B67 59 53 49
1B6A 43 41 4C
1B6D 20 44 43
1B70 56 49 43
1B73 45 3F 9B

```

```

1B76 55 1B      DEFDEV   . WORD  DEDE
1B7B 20 F1 21    JBR     GETLIN
1B7E 20 9E 22    JBR     GETFIL  GET LOGICAL DEVICE NAME
1B7E AD AE 14    LDA     PTR
1B81 C9 03      CMP     #3      SHOULD BE DEVICE NAME ONLY
1B83 D0 4C      BNE     B01     BAD SYNTAX
1B85 20 9E 22    JBR     GETFIL  GET PHYSICAL DEVICE
1B8B 20 79 22    JBR     PERX
1B8B AD 29 13    LDA     PAR     GET LOG DEV
1B8E 20 21 1C    JBR     HATS     SEARCH HAT (HANDLER ADDRESS TABLE)
1B91 10 0D      BPL     DD1     IF FOUND

```

```

; DIDNT FIND IT SO FIND EMPTY SLOT

```

```

1B93 A9 00      LDA     #0
1B95 20 21 1C    JBR     HATS
1B9B 30 22      BHI     STF     TABLE IS FILLED
1B9A AD 29 13    LDA     PAR
1B9D 9D 1A 03    STA     HATABB,X  PUT NAME IN TABLE
1BA0 A9 9B      DD1      LDA     #DRHL
1BA2 9D 1B 03    STA     HATABB+1,X
1BA5 A9 25      LDA     #DRHH
1BA7 9D 1C 03    STA     HATABB+2,X
1BAA BE AD 14    GTX     BAVX

```

```

; GET UP LDNT, PDST ENTRIES

```

```

1BAD AD 29 13    LDA     PAR     GET DEVICE
1BB0 20 2E 1C    JBR     FLDB
1BB3 10 40      BPL     B01     IF FOUND IT

```

COLLEEN DISK UTILITIES

```

1BB5 A7 00      LDA    #0
1BB7 20 2E 1C    JSR    FLDB
1BB8 10 33      BPL     BU2
;NO ROOM IN TABLE
1BBC A9 C6      STF     LDA    #TFL
1BDE A2 18      LDX     #TFH
1BC0 20 61 23    DSX     JSR    DBPLIN
1BC3 4C 91 16    JMP     MENUBL
1BC6 54 41 42    TF      BYTE  "TABLE FULL".CR
1BC9 4C 45 20
1BCC 46 55 4C
1BCF 4C 98

```

```

001B            TFH     =      TF/256
00C6            TFL     =      -256*TFH+TF
1BD1            B82
1BD1            B81

```

```

1BD1 A9 D7      LDA     #BSML
1BD3 A2 18      LDX     #BSMH
1BD5 D0 E9      BNE     DSX
1BD7 43 41 4E    BSM     BYTE  "CANT DEFINE THIS DEVICE".CR

```

```

1BDA 54 20 44
1BDD 45 46 49
1BE0 4E 45 20
1BE3 54 48 49
1BE6 53 20 44
1BE9 45 56 49
1DEC 43 45 98

```

```

001B            BSMH    =      BSM/256
00D7            BSMH    =      -256*BSMH+BSM
1BEF AD 29 13    BSMH    LDA     PAR
1BF2 9D 9D 25    BU2     STA     LDNT,X

```

```

1BF5 AD 28 13    ;SEE NOW IF THIS IS A DELETE
1BF8 C7 98      BUI     LDA     PAR+2

```

```

1BFA D0 0E      BNE     CRD      NO. GO CREATE DEVICE
;DELETE OPERATION, ZERO LDNT AND HAT ENTRIES

```

```

1BFC A9 00      LDA     #0
1BFE 9D 9D 25    STA     LDNT,X
1C01 AE AD 14    LDX     SAVX
1C04 9D 1A 03    STA     HATARS,X
1C07 4C 91 16    JMP     MENUBL

```

```

;CREATE DEVICE - HAT AND LDNT ARE SET UP, MAKE PDBT ENTRY
1C0A 0A          CRD     TXA

```

```

1C0B 0A          ASL     A
1C0C 0A          ASL     A
1C0D 0A          ASL     A
1C0E 0A          ASL     A

```

```

1C0F AA          TAX

```

```

1C10 A0 00      LDY     #0
1C12 B9 2C 13    CPBL     LDA     PAR+3,Y GET BYTE OF PHYSICAL DEV: FILSPEC
1C15 9D A8 25    STA     PDBT,X STORE IN TABLE

```

```

1C18 EB          INX
1C19 CB          INY
1C1A C0 10      CPY     #16
1C1C D0 E4      BNE     CPBL
1C1E 4C 91 16    JMP     MENUBL

```

```

;HATB FINDS DEVICE NAMED IN A, RETURNS ITS INDEX IN X

```

COLLEEN DISK UTILITIES

```

1C21 A2 21      HATB LDX  #MAXDEV
1C23 DD 1A 03    HATBL CMP  HATABS, X
1C24 F0 05      BEQ   HATX

```

```

1C2B CA         DEX
1C29 CA         DEX
1C2A CA         DEX

```

```

1C2B 10 F6      BPL   HATBL
1C2D 60         RTB

```

```

; FLDB IS THE SAME THING FOR LDNT

```

```

1C2E A2 04      FLDB LDX  #4
1C30 DD 9D 25    FLL   CMP  LDNT, X
1C33 F0 03      BEQ   FLDX

```

```

1C35 CA         DEX
1C36 10 F8      BPL   FLL
1C3B 60         FLDX RTB

```

```

; WBOOT - WRITE OUT NEW DOS.SYS FILE

```

```

1C39 8B 1C      WBOOT .WORD 0WMO
1C3B 20 33 22    JSR   CHROET
1C3E C9 B9      CMP   #'Y
1C40 D0 2A      BNE   WBX      EXIT UNLESS Y
1C42 A9 6F      LDA   #WBMOL
1C44 A2 1C      LDX   #WBMOLH

```

```

1C46 20 61 23    JSR   DBPLIN
1C49 A9 03      LDA   #OPEN
1C4B A2 10      LDX   ##10

```

```

1C4D 9D 42 03    STA   ICCOM, X
1C50 A9 A7      LDA   #DBL
1C52 9D 44 03    STA   ICBAL, X

```

```

1C55 A9 1C      LDA   #DBH
1C57 9D 45 03    STA   ICBAL, X
1C5A A9 08      LDA   #B

```

```

1C5C 9D 4A 03    STA   ICAX1, X
1C5F 20 C2 23    JSR   CIOCL
1C62 A2 10      LDX   ##10

```

```

1C64 A9 0C      LDA   #CLOSE
1C66 9D 42 03    STA   ICCOM, X
1C69 20 C2 23    JSR   CIOCL

```

```

1C6C 4C 91 16    WBX   JMP   MENUBL
1C6F 57 52 49    WRMO   .BYTE "WRITING NEW DOS.SYS FILE", CR
1C72 54 49 4E

```

```

1C75 47 20 4E
1C7B 4B 07 20
1C7B 44 4F 83

```

```

1C7E 2E 53 59
1C81 53 20 46
1C84 49 4C 45

```

```

1C87 9B

```

```

001C            WBMOLH HILO  WBMOL
006F            WBMOL  =     WBMOL/256
1C8B 54 59 50    WBMOL  =     -256*WBMOLH+WBMOL
1C8B 45 20 22    WBMOL  =     "TYPE ", 122, "Y", 122, " TO WRITE NEW DOS FILE", CR
1C8E 59 22 20
1C91 54 4F 20
1C94 57 52 49
1C97 54 45 20
1C9A 4F 45 57

```

COLLEEN DISK UTILITIES

```

1CA0 53 20 46
1CA3 49 4C 45
1CA6 9B
1CA7 44 3A 44      DS      .BYTE      "D: DOB. BYB", CR
1CAA 4F 53 2E
1CAD 53 59 53
1CB0 9B

```

```

001C      HIL0      DB
00A7      DSL      =      DB/256
           =      -256*DSH+DB

```

LDFIL - LOAD FILE FUNCTION

```

1CB1 F3 1C      LDFIL      .WORD      LFM0
1CB3 A9 91      LDA      @MNSLL
1CB5 8D E0 02      STA      RUNAD      GET RUN ADDRESS TO MENU SELECT
1CB8 A9 16      LDA      @MNSLH      SO IF NONE IN FILE WE GO THERE
1CBA 8D E1 02      STA      RUNAD+1
1CBD 20 84 22      JBR      GETIC1
1CC0 20 79 22      JBR      PERX
1CC3 20 19 13      JBR      LOADV
1CC6 E0 00      CPX      #0      PROCESS LOAD SUBR RESPONSE
1CCB D0 03      BNE      ++5      BR IF ERRORB
1CCA 6C E0 02      JMP      (RUNAD) IF OK, GO TO RUN ADDRESS
1CCD E0 03      CPX      #3
1CCF F0 07      BEQ      NLF      IF BAD LOAD FILE
1CD1 9B      TYA      OTHERWISE WE GOT A CIO ERROR
1CD2 4C CA 23      JMP      CIDER      GO SAY WHAT IT IS
1CD5 4C 91 16      LDFX      JMP      MENUSL
1CDB A9 E5      NLF      LDA      @BLFL
1CDA A2 1C      LDY      @BLFH
1CDC 20 61 23      JSR      DBPLIN      BAD LOAD FILE MSG
1CDF 20 F8 23      JSR      CLOSX      CLOSE THE FILE
1CE2 4C 91 16      JMP      MENUSL      EXIT
1CE5 42 41 44      BLF      .BYTE      "BAD LOAD FILE", CR
1CE8 20 4C 4F
1CEB 41 44 20
1CEE 46 49 4C
1CF1 45 9B

```

```

001C      HIL0      BLF
00E5      BLFH      =      BLF/256
           BLFL      =      -256*BLFH+BLF
1CF3 4C 4F 41      LFM0      .BYTE      "LOAD FROM WHAT FILE?", CR

```

```

1CF6 44 20 46
1CF9 52 4F 4D
1CFC 20 57 4B
1CFF 41 54 20
1D02 46 49 4C
1D05 45 3F 9B

```

LKFILE - DO CIO LOCK

```

1D08 1D 1D      LKFIL      .WORD      LKM0
1D0A 20 84 22      JSR      GETIC1
1D0D 20 79 22      JSR      PERX
1D10 A9 23      LDA      #LOCK
1D12 A2 10      LDY      #610

```

COLLEEN DISK UTILITIES

1D17 20 C2 23		JSR	CIOCL
1D1A 4C 91 16		JMP	MENUBL
1D1D 57 4B 41	ULMO	. BYTE	"WHAT FILE TO LOCK?", CR

1D20 54 20 46
1D23 49 4C 45
1D26 20 54 4F
1D29 20 4C 4F
1D2C 43 4B 3F
1D2F 9B

; ULFIL - DO CIO UNLOCK
ULFIL . WORD ULMO

1D30 45 1D			
1D32 20 84 22		JSR	QETIC1
1D35 20 79 22		JSR	PERX
1D38 A9 24		LDA	#UNLOCK
1D3A A2 10		LDX	#10
1D3C 9D 42 03		STA	ICCOM, X
1D3F 20 C2 23		JSR	CIOCL

1D42 4C 91 16		JMP	MENUBL
1D45 57 4B 41	ULMO	. BYTE	"WHAT FILE TO UNLOCK?", CR

1D48 54 20 46
1D4B 49 4C 45
1D4E 20 54 4F
1D51 20 55 4E
1D54 4C 4F 43
1D57 4B 3F 9B

COLLEEN DISK UTILITIES

1D5A 44 55 50 DIMO . BYTE "DUP DISK-SOURCE, DEST DRIVEB?", CR

1D5D 20 44 49

1D60 53 48 2D

1D63 53 4F 55

1D66 52 43 45

1D69 2C 44 45

1D6C 53 54 20

1D6F 44 52 49

1D72 56 45 53

1D75 3F 9B

1D77 54 59 50 OK . BYTE "TYPE ", \$22, "Y", \$22, " IF OK TO USE PROGRAM AREA", CR

1D7A 45 20 22

1D7D 59 22 20

1D80 49 46 20

1D83 4F 4B 20

1D86 54 4F 20

1D89 55 53 45

1D8C 20 50 52

1D8F 4F 47 52

1D92 41 4D 20

1D95 41 52 45

1D98 41 9B

001D

OKH

HILO

OK

OK/256

0077

OKL

"

-256*OKH+OK

DUPLICATE DISK ROUTINE

RVTOC READS VOLUME TABLE OF CONTENTS SECTOR

1D9A A9 01

RVTOC

LDA

#1

1D9C BD 0B 03

STA

DSHI

READ VTOC SECTOR

1D9F A9 6B

LDA

##6B

1DA1 BD 0A 03

STA

DBLO

1DA4 A9 14

LDA

#DB1H

1DA6 BD 05 03

STA

DBUFH

1DA9 A9 21

LDA

#DB1L

1DAB BD 04 03

STA

DBUFLO POINT DCR AT PAR

1DAE 20 C5 1F

JSR

RBEC1

1DB1 A9 00

LDA

#0

1DB3 BD AE 14

STA

PTR

1DB6 AD 33 13

LDA

PAR+9A

1DB9 BD AB 14

STA

CBRC

BYTE OF ALLOC MAP

1DUC A9 0B

LDA

#B

1DBE BD AF 14

STA

IPTR

COUNT BITS IN BYTE

1DC1 A9 00

LDA

#0

1DC3 BD 0B 03

STA

DSHI

POINT T SECTOR ONE

1DC6 A9 01

LDA

#1

1DC9 BD 0A 03

STA

DBLO

1DCB 60

RIS

COLLEEN DISK UTILITIES

```

1DCE 20 F1 21      JSR   GETLIN
1DD1 20 53 25      JSR   GETDM
1DD4 8D A3 14      STA   UNNO      UNIT NO FOR READ
1DD7 20 53 25      JSR   GETDM
1DDA 8D AC 14      STA   CDEB
1DDD 20 79 22      JSR   PERX
1DE0 AD A3 14      LDA   UNNO
1DE3 CD AC 14      CMP   CDEB      IF BOTH UNITS THE SAME
1DE6 F0 45      BEQ   BDD      SINGLE DRIVE DUP
1DEB A2 1E      LDX   #IBDH
1DEA A9 0E      LDA   #IBDL
1DEC 20 61 23      JSR   DSPLIN  PROMPT TO INSERT BOTH DISKS
1DEF 20 33 22      JSR   CHROET
1DF2 20 9A 1D      JSR   RVTOC
1DF5 20 87 1F      NSBC      JSR   AAM      ADV ALLOC MAP ONE BIT
1DF8 30 0C      BMI   CEND
1DFA 20 BB 1F      JSR   RBEC
1DFD AD AC 14      LDA   CDEB      UNIT TO WRITE TO
1E00 8D 01 03      STA   DUNIT
1E03 20 D0 1F      JSR   DKWRT
1E06 20 A4 1F      CEND      JSR   ABP
1E09 D0 EA      BNE   NSBC      IF MORE SECTORS
1E0B 4C 91 16      JMP   MENUCL  DONE
1E0E 49 4E 53      IBDD      .BYTE  "INSERT BOTH DISKS, TYPE RETURN".CR
1E11 45 52 54
1E14 20 42 4F
1E17 54 48 20
1E1A 44 49 53
1E1D 48 53 2C
1E20 20 54 59
1E23 50 45 20
1E26 52 45 54
1E29 55 52 4E
1E2C 9B

```

```

001E      HILO      IBDD
000E      IBDD      =      IBDD/256
          IBDL      =      -256*IBDH+IBDD
          /SINGLE DRIVE DISK DUP

```

/FIND AVAIL MEMORY. FILL FROM SOURCE DISK, SWAP, EMPTY, SWAP, REPEAT.

```

1E2D AD E7 02      BDD      LDA      MEMLO      END OF DUP
1E30 8D B4 14      STA      STVEC
1E33 AD E8 02      LDA      MEMLO+1
1E36 8D B5 14      STA      STVEC+1
1E39 AD E5 02      LDA      MEMTOP
1E3C 3B      SEC
1E3D E7 80      BDC      #180      END-128 TO SEE IF ROOM
1E3F 8D B2 14      STA      T1      FOR ANOTHER SECTOR
1E42 AD E6 02      LDA      MEMTOP+1
1E45 E7 00      BDC      #0
1E47 8D B3 14      STA      T1+1
          /SEE IF ROOM FOR AT LEAST ONE SECTOR!
1E4A AD B3 14      LDA      T1+1
1E4D CD B5 14      CMP      STVEC+1
1E50 30 0A      BMI      NORM
1E52 D0 12      BNE      ENUF
1E54 AD B2 14      LDA      T1
1E57 CD B4 14      CMP      STVEC

```

CILLEN DISK UTILITIES

```

IE5C A9 34      NORM   LDA    #NRML
IE5E A2 1F      LDX    #NRMH
IE60 20 61 23   JSR     DBPLIN
IE63 4C 91 16   JMP     MENUBL
IE66 20 F2 1F   ENUF    JSR     CKMEM    SEE IF OK TO USE USER AREA
IE69 A9 00      LDA    #0
IE6B 8D B1 14   STA    OPT      SET UP FOR READ HERE FIRST PASS
IE6E A9 44      LDA    #ISDL
IE70 A2 1F      LDX    #ISDH
IE72 20 61 23   JSR     DBPLIN
IE75 20 33 22   JSR     CHROET
IE7B 20 9A 1D   JSR     RVTOT    READ VTOC AFTER 1ST INSERT SOURCE MBO
IE7B AD 0A 03   LDA    DBLO    COPY INITIAL WRITE POINTERS
IE7E 8D A6 14   STA    BWDP    TO INITIAL READ POINTERS
IE81 AD 0B 03   LDA    DBHI
IE84 8D A7 14   STA    BWDP+1
IE87 AD AE 14   LDA    PTR
IE8A 8D AB 14   STA    BWDP+2
IE8D AD AF 14   LDA    IPTR
IE90 8D A9 14   STA    BWDP+3
IE93 AD AB 14   LDA    CBRC
IE96 8D AA 14   STA    BWDP+4
IE99 4C AB 1E   JMP     LRS1    SKIP FIRST READ PROMPT

```

READ FROM SOURCE DISK TIL BUF FULL OR END OF DATA.

```

IE9C A9 00      DORD    LDA    #0      FLAG WE ARE READING
IE9E 8D B1 14   STA    OPT
IEA1 A9 44      LDA    #ISDL    INSERT SRC DISK
IEA3 A2 1F      LDX    #ISDH
IEA5 20 61 23   XBLK    JSR     DBPLIN
IEA8 20 33 22   JSR     CHROET

```

```

IEAB 20 00 1F   LRS1    JSR     DOBWDP    SWAP SECTOR AND BITMAP POINTERS

```

LOOP READING/WITING SECTORS TO BUFFER AREA

```

IEAE 20 87 1F   LRS    JSR     AAM      ADVANCE ALLOCATION MAP
IEB1 30 1F      BMI     ABPT     IF FREE, ADV SECTOR POINTER AND TRY AGAIN
IEB3 2C B1 14   BIT     OPT      SEE WHAT MODE
IEB6 30 06      BMI     DOW     OR IF WRITE
IEBB 20 C5 1F   JSR     RBEC1    DO READ
IEBB 4C C1 1E   JMP     IOD
IEBE 20 D0 1F   DOW     JSR     DKWRT  DO WRITE
IEC1 AD 04 03   IOD     LDA    DBUFLO  ADVANCE BUFFER POINTER
IEC4 1B        CLC

```

```

IEC5 69 80      ADC     #80
IEC7 8D 04 03   STA    DBUFLO
IECA AD 03 03   LDA    DBUFHI
IECD 69 00      ADC     #0
IECF 8D 05 03   STA    DBUFHI

```

```

IED2 20 A4 1F   ASPT    JSR     ASP      GO ADVANCE SECTOR POINTER
IED5 F0 21      BEQ     STDD1    ALL SECTORS DONE, SWAP TO DEST DISK
IED7 AD 83 14   LDA    T1+1    SEE IF ROOM FOR ANOTHER
IEDA CD 05 03   CMP     DBUFHI
IEDD 30 0A      BMI     STDD      IF NO ROOM, GO SWAP
IEDF D0 CD      BNE     LRS      ROOM SO CONTINUE
IEE1 AD B2 14   LDA    T1
IEE4 CD 04 03   CMP     DBUFLO
IEE7 10 C5      BPL     LRS      ROOM.

```

SWAP DISKS AND CONTINUE

```

IEE9 AD B1 14   STDD    LDA    OPT
IEEC 30 AE      BMI     DORD    IF WAS WRITE, GO READ

```

COLLEEN DISK UTILITIES

1EF1 A9 63	LDA	#IDDL	INSERT DEST DISK
1EF3 A2 1F	LDX	#IDDH	
1EF9 4C A5 1E	JMP	XBLK	GO DO WRITE
1EFB AD B1 14	BTDD1	LDA	OPT END OF DATA
1EFB 10 F1	BPL	BTDD2	IF READ GO WRITE
1EFD 4C 91 16	JMP	MENUSL	IF WRITE WE ARE DONE

DOBDWP - EXCHANGE CURRENT AND SAVED BITMAP&SECTOR POINTERS
 ALSO INIT BUFFER POINTER

1F00 A0 04	DOBDWP	LDY	#4
1F02 B9 2A 1F	BWLOH	LDA	BWATL, Y
1F05 B5 1A	STA	RAMLO	
1F07 B9 2F 1F	LDA	BWATH, Y	
1F0A B5 1B	STA	RAMLO+1	GET ADDRESS FROM TABLE TO RAMLO
1F0C A2 00	LDX	#0	
1F0E A1 1A	LDA	(RAMLO, X)	GET WHATS THERE
1F10 4B	PHA		
1F11 B9 A6 14	LDA	BWDP, Y	
1F14 B1 1A	STA	(RAMLO, X)	
1F16 6B	PLA		
1F17 99 A6 14	STA	BWDP, Y	
1F1A 0B	DEY		
1F1B 10 E5	BPL	BWLOP	
1F1D AD B4 14	LDA	STVEC	
1F20 BD 04 03	STA	DBUFLO	
1F23 AD B5 14	LDA	STVEC+1	
1F26 BD 05 03	STA	DBUFHI	
1F29 60	RTS		

WHAT A MESS

0003	DBLOH	HILO	DBLO
000A	DBLOL	=	DBLO/256
		=	-256*DBLOH+DBLO
0003	DBHIH	HILO	DBHI
000B	DBHIL	=	DBHI/256
		=	-256*DBHIH+DBHI
0014	PTRH	HILO	PTR
00AE	PTRL	=	PTR/256
		=	-256*PTRH+PTR
0014	IPTRH	HILO	IPTR
00AF	IPTRL	=	IPTR/256
		=	-256*IPTRH+IPTR
0014	CBRCH	HILO	CBRC
00AB	CBRCL	=	CBRC/256
		=	-256*CBRCH+CBRC

1F2A 0A 0B AE	BWATL	. BYTE	DBLOL, DBHIL, PTRL, IPTRL, CBRCL
1F2D AE AB			
1F2F 03 03 14	BWATH	. BYTE	DBLOH, DBHIH, PTRH, IPTRH, CBRCH
1F32 14 14			

1F34 4E 4F 54	NRH	. BYTE	"NOT ENOUGH ROOM", CR
---------------	-----	--------	-----------------------

1F37 20 45 4E			
1F3A 4E 55 47			
1F3D 4B 20 52			
1F40 4F 4F 4D			
1F43 9B			
1F44 49 4E 53	IRD	. BYTE	"INSERT SOURCE DISK, TYPE RETURN", CR
1F47 45 52 54			

COLLEEN DISK UTILITIES

1F4D 55 52 43
 1F50 45 20 44
 1F53 49 53 4B
 1F56 2C 54 39
 1F59 50 45 20
 1F5C 52 45 54

1F5F 55 52 4E

1F62 9B

1F63 49 4E 53

IDD . BYTE "INSERT DESTINATION DISK TYPE RETURN", CR

1F66 45 52 54

1F69 20 44 45

1F6C 53 54 49

1F6F 4E 41 54

1F72 49 4F 4E

1F75 20 44 49

1F78 53 4B 2C

1F7B 54 59 30

1F7E 45 20 52

1F81 45 54 55

1F84 52 4E 9B

001F

NRMH

HILO

NRH

0034

NRML

=

NRH/256

-256*NRMH+NRH

HILO

IBD

001F

IBDH

=

IBD/256

0044

IBDL

=

-256*IBDH+IBD

HILO

IDD

001F

IDDH

=

IDD/256

0063

IDDL

=

-256*IDDH+IDD

AAH - ADVANCE ALLOCATION MAP ONE BIT.

RETURN MINUS IF FREE.

1F87 0E AB 14

AAH

ABL

CBRC

NEXT BIT OF ALLOC MAP

1F8A CE AF 14

DEC

IPTR

1F8D 00 11

BNE

CBIT

IF DONE WITH THIS BYTE

1F8F EE AE 14

INC

PTR

GET NEXT ONE

1F92 AE AE 14

LDX

PTR

1F95 BD 2B 14

LDA

DB1+AA,X

1F98 BD AB 14

STA

CBRC

1F9B A9 0B

LDA

9B

1F9D BD AF 14

STA

IPTR

1FA0 AD AB 14

CBIT

LDA

CBRC

CHECK THE BIT

1FA3 60

RTS

ABP - ADVANCE SECTOR POINTER IN DCB.

RETURN EQ IF AT END.

1FA4 AD 0A 03

ASP

LDA

DBLO

SEE IF END

1FA7 C9 CF

CMF

#207

1FA9 D0 07

BNE

NXB

1FAB AD 0B 03

LDA

DBHI

1FAE C9 02

CMF

#2

1FB0 F0 0B

BEG

ABPX

ALL DONE

1FB2 EE 0A 03

NXB

INC

DBLO

1FB5 D0 03

BNE

ABPX

1FB7 EE 0B 03

INC

DBHI

1FBA 60

ABPX

RTS

RREC - READ A SECTOR WHOSE NUMBER IS IN DCB

1FBB A9 A1

RREC

LDA

#DBUFL

1FBD BD 04 03

STA

DBUFLD

COLLEEN DISK UTILITIES

1FC2 8D 05 03		STA	DBUFHI	
1FC5 AD A3 14	RBEC1	LDA	UNNO	
1FCB 8D 01 03		STA	DUNIT	
1FCB A9 52		LDA	#52	GET SECTOR
1FCD 4C D2 1F		JMP	CLDKH	

DKWRT - WRITE A SECTOR, NUMBER IN DCB

1FD0 A9 57	DKWRT	LDA	#57	
1FD2 8D 02 03	CLDKH	STA	DCOMND	
1FD5 A9 02		LDA	#2	SET RETRY COUNT
1FD7 8D A4 14		STA	RCNT	

1FDA 20 53 E4	CLD1	JSR	DKIND	
1FDD AD 03 03		LDA	DSTATB	
1FE0 C9 01		CMF	#1	
1FE2 F0 08		BEO	DRTB	OK STATUS
1FE4 CE A4 14		DEC	RCNT	
1FE7 10 F1		DPL	CLD1	
1FE9 4C CA 23		JMP	CIOER	CIO ERROR, GO SAY WHICH
1FEC 60	DRTB	RTB		

CKMEM - ASK IF OK TO USE USER AREA

1FED 68	DDXT	PLA		POP RETURN ADDRESS
1FEE 68		PLA		
1FEF 4C 91 16		JMP	MENUSL	
1FF2 A3 08	CKMEM	LDA	WARMST	IF MEMORY WAS INTACT
1FF4 F0 12		BEO	CPTR1	QUERY TO BOMB IT
1FF5 A9 77		LDA	#OKL	
1FF8 A2 1D		LDX	#OKH	
1FFA 20 61 23		JSR	DSPLIN	
1FFD 20 33 22		JSR	CHROET	
2000 C9 59		CMF	#Y	
2002 D0 E9		BNE	DDXT	
2004 A9 00		LDA	#0	
2006 85 08		STA	WARMST	
2008 60	CPTR1	RTB		

COLLEEN DISK UTILITIES

DUFFIL - COPY FILE FROM ONE DISK TO ANOTHER

2009 4E 41 4D	DWFM	. BYTE	"NAME OF FILE TO MOVE?".CR
200C 45 20 4F			
200F 46 20 46			
2012 49 4C 45			
2015 20 54 4F			
2018 20 4D 4F			
201B 56 45 3F			
201E 98			
201F 09 20	DUFFIL	. WORD	DPFM
2021 20 84 22	JSR	GETIC1	GET A DEV:FILESPEC
2024 20 79 22	JSR	PERX	CHECK FOR ERRORB
2027 20 F2 1F	JSR	CKMEM	
202A A2 10	LDX	#*10	
202C A9 03	LDA	#OPEN	
202E 9D 42 03	STA	ICCOM.X	
2031 A9 04	LDA	#4	
2033 9D 4A 03	STA	ICAX1.X	
2036 20 C2 23	JSR	CIOCL	
2039 AD E5 02	LDA	MENTOP	SET BUFFER LENGTH
203C 3B	SBC		TO ALL AVAILABLE MEMORY
203D ED E7 02	SBC	MEMLO	
2040 9D 48 03	STA	ICBLL.X	
2043 AD E6 02	LDA	MENTOP+1	
2046 ED E8 02	SBC	MEMLO+1	
2049 9D 49 03	STA	ICBLH.X	
204C AD E7 02	LDA	MEMLO	
204F 9D 44 03	STA	ICBAL.X	
2052 AD E8 02	LDA	MEMLO+1	
2055 9D 45 03	STA	ICBAH.X	
2058 A9 06	LDA	#GETCHR	
205A 9D 42 03	STA	ICCOM.X	
205D 20 56 E4	JSR	CIO	
2060 C0 88	CPY	#*88	
2062 F0 18	BEG	OKB	IF EOF, WE READ WHOLE FILE SO OK
2064 C0 01	CPY	#1	
2066 F0 03	BEG	#+5	
2068 4C C9 23	JMP	CIOER1	NOT OK OR EOF SO GIVE ERROR MESSAGE
206B A9 34	LDA	#NRML	OK, SAY NO ROOM
206D A2 1F	LDX	#NRMH	
206F 20 61 23	JSR	DBPLIN	
2072 A9 0C	LDA	#CLOSE	
2074 A2 10	LDX	#*10	
2076 9D 42 03	STA	ICCOM.X	
2079 20 56 E4	JSR	CIO	
207C 4C 91 16	JMP	MENUSL	
207F A9 0C	LDA	#CLOSE	
2081 A2 10	LDX	#*10	
2083 9D 42 03	STA	ICCOM.X	
2086 BD 48 03	LDA	ICBLL.X	
2089 BD 82 14	STA	T1	
208C BD 49 03	LDA	ICBLH.X	
208F BD 83 14	STA	T1+1	
2092 20 56 E4	JSR	CIO	
2095 A2 1F	LDX	#IDDH	
2097 A9 63	LDA	#IDDL	
2099 20 61 23	JSR	DBPLIN	SAY TO SWAP DISKS

VF-141Z
VLS-1410

COLLEEN DISK UTILITIES

```

209F A2 10      LDX    #*10
20A1 A9 29      LDA    #PARL
20A3 9D 44 03   BTA    ICBAL, X POINT TO FILENAME AGAIN
20A6 A9 13      LDA    #PARH
20A8 9D 45 03   BTA    ICBAL, X
20AB A9 03      LDA    #OPEN
20AD 9D 42 03   BTA    ICCOM, X
20B0 A9 08      LDA    #B
20B2 9D 4A 03   BTA    ICAX1, X
20B5 20 C2 23   JSR    CIOCL OPEN FOR OUTPUT
20B8 AD E7 02   LDA    MEMLO
20BB 9D 44 03   BTA    ICBAL, X
20BE AD E8 02   LDA    MEMLO+1 SET UP ADR AND LENGTH
20C1 9D 45 03   BTA    ICBAL, X
20C4 AD B2 14   LDA    T1
20C7 9D 48 03   BTA    ICBLL, X
20CA AD B3 14   LDA    T1+1
20CD 9D 49 03   BTA    ICBLL, X
20D0 A9 0A      LDA    #PUTCHR
20D2 9D 42 03   BTA    ICCOM, X
20D5 20 C2 23   JSR    CIOCL DO THE WRITE
20D8 4C 72 20   JMP     DFCX CLOSE AND EXIT

```

SAVE FILE ROUTINE

```

20DB A0 21      SAVFIL .WORD BFM0
20DD 20 B4 22   JSR    GETIC1
20E0 AD B1 14   LDA    OPT
20E3 48         PHA
20E4 AE AE 14   LDX    PTR PUT EOL ON FILENAME
20E7 A9 9B      LDA    #CR
20E9 9D 28 13   STA    PAR-1, X
20EC 20 BA 24   JSR    GETNO GET HEX PARAMETER
20EF BD BB 14   BTA    LDST
20F2 BE BC 14   BTX    LDST+1
20F5 20 BA 24   JSR    GETNO END ADDRESS
20F8 BD BD 14   BTA    LDND
20FB BE BE 14   BTX    LDND+1
20FE 2C A2 14   BIT    PER SEE IF PARAM ERRORS
2101 10 03      BPL    ++5
2103 4C 91 14   JMP     MENUCL IF ERRORS
2106 A9 00      LDA    #0
2108 BD B1 14   BTA    OPT
210B 68         PLA OPTION CHAR FROM FILENAME
210C C9 41      CMV    #'A IF APPEND
210E D0 03      BNE    ++5
2110 CE B1 14   DEC     OPT SET DT=OFF

```

OPEN THE FILE

```

2113 A2 10      LDX    #*10
2115 A9 03      LDA    #OPEN
2117 9D 42 03   BTA    ICCOM, X
211A 2C B1 14   BIT    OPT IF APPEND
211D 30 04      BMI    ++6
211F A9 08      LDA    #B
2121 D0 02      BNE    ++4
2123 A9 09      LDA    #9

```

COLLEEN DISK UTILITIES

```

2128 20 C2 23      JSR    CIOCL
                   I WRITE SAVE FILE HEADER
2128 A2 10          LDX    #*10
212D A9 0A          LDA    #PUTCHR
212F 9D 42 03      STA    ICCOM, X
2132 2C B1 14      BIT    OPT      IF APPEND,
2135 30 17          BMT    WDR      DONT WRITE ANOTHER HEADER
2137 A9 BF          LDA    #BAVHL
2139 9D 44 03      STA    ICBAL, X
213C A9 14          LDA    #BAVHH
213E 9D 45 03      STA    ICBALH, X
2141 A9 02          LDA    #2
2143 9D 48 03      STA    ICBLL, X
2146 A9 00          LDA    #0
2148 9D 49 03      STA    ICBLLH, X
2148 20 C2 23      JSR    CIOCL
                   I WRITE DATA RECORD
214E A2 10          WDR LDX    #*10
2150 A9 04          LDA    #4
2152 9D 48 03      STA    ICBLL, X
2155 A9 00          LDA    #0
2157 9D 49 03      STA    ICBLLH, X
215A A9 BB          LDA    #DTHL DATA RECORD HEADER
215C 9D 44 03      STA    ICBAL, X
215F A9 14          LDA    #DTHH
2161 9D 45 03      STA    ICBALH, X
2164 20 C2 23      JSR    CIOCL NO. WRITE DATA REC HDR
2167 A2 10          LDX    #*10
2169 38            BEC
216A AD B0 14      LDA    LDND
216D ED B8 14      SBC    LDBT
2170 9D 48 03      STA    ICBLL, X
2173 AD BE 14      LDA    LDND+1
2176 ED BC 14      SBC    LDBT+1
2179 9D 49 03      STA    ICBLLH, X
217C FE 48 03      INC    ICBLL, X
217F D0 03          BNE    ++5
2181 FE 49 03      INC    ICBLLH, X
2184 AD B9 14      LDA    LDBT
2187 9D 44 03      STA    ICBAL, X
218A AD BC 14      LDA    LDBT+1
218D 9D 45 03      STA    ICBALH, X
2190 20 C2 23      JSR    CIOCL WRITE THE DATA
2193 A2 10          LDX    #*10
2195 A9 0C          LDA    #CLOSE
2197 9D 42 03      STA    ICCOM, X
219A 20 C2 23      JSR    CIOCL
219D 4C 91 16      JMP    MENU8L DONE.
21A0 53 41 56      BFMD . BYTE "SAVE--GIVE FILE, START, END", CR
21A3 45 2D 2D
21A6 47 49 56
21A9 45 20 46
21AC 47 4C 45
21AF 2C 20 53
21B2 54 41 52
21B5 54 2C 20
21B8 45 4E 44

```

COLLEEN DISK UTILITIES

SUBROUTINES

INITIO - CLOSE ANY OPEN IOCB

```

21BC A2 70      INITIO LDX    #070
21BE BD 40 03    IIOI  LDA    ICHID,X
21C1 C9 FF      CMP    #0FF    CHECK FOR CLOSED IOCB
21C3 F0 08      RFG    NXIOCB  ALREADY CLOSED, CHECK NEXT IOCB
21C5 A9 0C      LDA    #CLOSE
21C7 9D 42 03    BTB    ICCOM,X
21CA 20 56 E4    JSR    CIO

```

```

21CD BA      NXIOCB TXA
21CE 38      SEC
21CF E9 10    SBC    #010

```

```

21D1 AA      TAX
21D2 10 EA    BPL    IIOI
21D4 A2 00    LDX    #0      REOPEN IOCB TO E:

```

```

21D6 A9 03    LDA    #OPEN
21D8 BD 42 03 BTB    ICCOM
21DB A9 EE    LDA    #ECL

```

```

21DD BD 44 03 BTB    ICBAL
21E0 A9 21    LDA    #ECH
21E2 BD 45 03 BTB    ICBAL

```

```

21E5 A9 0C    LDA    #0C
21E7 9D 4A 03 STA    ICAX1,X
21EA 20 56 E4 JSR    CIO

```

```

21ED 60      RTB
21EE 45 3A 9B EC    .BYTE    "E: ",CR
                HILO    EC

```

```

0021      ECH    =    EC/256
00EE      ECL    =    -256*ECH+EC

```

```

21F1 A9 9B      GETLIN LDA    #CR
21F3 A2 4F      LDX    #79
21F5 9D 51 13    STA    LINE,X
21F8 CA      DEX
21F9 10 FA      BPL    *-4

```

```

21FB A9 00      LDA    #0
21FD BD AE 14    STA    PTR
2200 BD AF 14    STA    IPTR
2203 BD A2 14    STA    PER
2206 20 0D 22    JSR    CIOGET
2209 20 0E 23    JSR    BCRDL
220C 60      RTB

```

CIOGET - GET LINE OF INPUT FROM SCREEN EDITOR

```

220D A9 05      CIOGET LIA    #0ETCRD+TEXT
220F BD 42 03    STA    ICCOM  SCREEN EDIT IOCB
2212 A9 51      LDA    #LBUFL
2215 BD 44 03    STA    ICBAL
2217 A9 13      LIA    #LBUFH

```

COLLEEN DISK UTILITIES

```

221C A7 50      LDA    #80
221E BD 48 03    STA    ICBLL
2221 A7 00      LDA    #0
2223 BD 49 03    STA    ICBLLH
2226 A2 00      LDX    #0
2228 20 56 E4    JSR    CIO      READ RECORD FROM SCREEN EDITOR
222B C0 80      CPY    #80      CHECK FOR BREAK ABORT STATUS
222D 00 03      BNE    **5
222F CE A2 14    DEC    PER      PARAM ERROR FLAG IS SET IF 80
2232 60          RTB

```

CHRGET - GET 1 CHAR FROM EDITOR IN A.

```

2233 A7 00      CHRGET LDA    #0
2235 BD A2 14    STA    PER
2238 20 0D 22    CHRG1 JSR    CIOGET GET A LINE FROM E:
223B AD 48 03    LDA    ICBLL HAVE CHAR COUNT
223E BD A4 14    STA    RCNT
2241 20 BE 23    JSR    BCROL
2244 AD A2 14    LDA    PER
2247 10 06      BPL    CHRG2 IF BREAK, CLOSE AND EXIT
2249 20 F8 23    JSR    CLOSX
224C 4C 91 16    JMP    MENU8L
224F AD A4 14    CHRG2 LDA    RCNT EXPECT 1 OR 2 CHARACTERS
2252 C9 03      CMP    #3
2254 30 0A      BMI    CHRG3 IF OK
2256 A7 64      LDA    #0LL
2258 A2 22      LDX    #0LH
225A 20 61 23    JSR    DRPLIN
225D 4C 38 22    JMP    CHRG1 TRY AGAIN
2260 AD 51 13    CHRG3 LDA    LINE GET 1ST CHAR
2263 60          RTB
2264 50 4C 45    OL      .BYTE "PLEASE TYPE 1 LETTER",CR
2267 41 53 45
226A 20 54 59
226D 50 45 20
2270 31 20 40
2273 45 54 84
2276 45 52 98

```

```

0022      OLH    HILO    OL
0064      OLL    -      OL/256
                -256*OLH+OL

```

PERX - EXIT IF PARAMETER ERRORS

```

2279 2C A2 14    PERX   BIT    PER
227C 30 01      BMI    PERX1
227E 60          RTB
227F 68          PERX1  PLA
2280 68          PLA
2281 4C 91 16    JMP    MENU8L

```

CH101 - READ LINE, GET FILENAME, POINT TO IT IN IOCB1

COLLEEN DISK UTILITIES

```

2290 A9 08      GETNAME LDA    #0
2292 BD 80 14   STA    CTR
2295 AC AE 14   LDY    PTR
2298 AE AF 14   LDX    IPTR
229B 4C F4 22   JMP     CFTE

```

```

      GET FILESPEC FROM INPUT LINE
229E AC AE 14   GETFIL LDY    PTR
22A1 AE AF 14   LDX    IPTR

```

```

22A4 A9 0A      LDA    #10
22A6 BD 80 14   STA    CTR

```

```

      AVOID GETTING JUNK ON VERY SHORT PARAMS

```

```

22A9 BD 51 13   LDA    LINE,X
22AC C9 2C      CMP     #'
22AE F0 3B      BEQ     ADDC
22B0 C9 9B      CMP     #CR
22B2 F0 34      BEQ     ADDC

```

```

22B4 BD 52 13   LDA    LINE+1,X
22B7 C9 2C      CMP     #'
22B9 F0 22      BEQ     QT1
22BB C9 9B      CMP     #CR

```

```

22BD F0 1E      BEQ     QT1
22BF A9 3A      LDA     #'      LOOK FOR : IN FILESPEC
22C1 DD 93 13   CMP     LINE+2,X  SEE IF HAVE COMPLETE FILESPEC ALREADY
22C4 F0 2E      BEQ     CFTE
22C6 DD 52 13   CMP     LINE+1,X
22C9 D0 12      BNE     QT1

```

```

22CB CE 80 14   DEC     CTR
22CE BD 51 13   LDA     LINE,X
22D1 C9 41      CMP     #'A
22D3 10 1F      BPL     CFTE      HAVE X:FILE, COMPLETE FILESPEC

```

```

      IF FALLS THRU, IS UNIT:FILE, ADD D

```

```

22D5 A9 44      QT2     LDA     #'D
22D7 99 29 13   STA     PAR,Y
22DA CB         JNY
22DB 10 17      BPL     CFTE

```

```

22DD CE 80 14   QT1     DEC     CTR
22E0 DD 51 13   CMP     LINE,X  AN UNLIKELY CASE (:FILE)
22E3 F0 F0      BEQ     QT2     TREAT :FILE AS U:FILE

```

```

22E5 CE 80 14   DEC     CTR
22E8 A9 44      ADDC     LDA     #'D
22EA 99 29 13   STA     PAR,Y

```

```

22ED CB         JNY
22EE A9 3A      LDA     #'
22F0 99 29 13   STA     PAR,Y

```

```

22F3 CB         JNY
22F4 A9 00      CFTE     LDA     #0
22F6 BD 81 14   STA     OPT
22F9 BD 51 13   CFTE1    LDA     LINE,X
22FC 99 29 13   STA     PAR,Y

```

```

22FF EB         INX
2300 CB         JNY
2301 C9 9B      CMP     #CR      LOOK FOR TERMINATOR

```

```

2303 F0 2C      RFO     EDC
      05 C9 2C      CMP     #'
2307 F0 2B      RFO     EDC

```

CMLFEN DISK UTILITIES

```

230B F0 2B      BEQ      POPT
230D C9 2E      CMP      #'      LOOK FOR START OF .EXT
230F D0 05      BNE      CFTE2
2311 A9 04      LDA      #4      FOUND, 4 MORE CHARS MAX
2313 BD 80 14    STA      CTR
2316 CE 80 14    CFTE2 DEC      CTR
2319 10 DE      BPL      CFTE1

```

GETB HERE IF TOO MANY CHARS IN FILENAME

```

231B A9 4B      LDA      #NTLL
231D A2 23      LDX      #NTLH
231F 20 61 23    JBR      DBPLIN NAME TOO LONG
2322 CE A2 14    DEC      PER      SET PARAMETER ERROR FLAG
2325 BD 51 13    BTE      LDA      LINE,X SKIP TO END

```

```

2328 EB          INX
2329 C9 2C      CMP      #'
232B F0 04      BEQ      EDC
232D C9 9B      CMP      #CR
232F D0 F4      BNE      BTE
2331 BE AF 14    EDC      STX      IPTR
2334 BC AE 14    STY      PTR
2337 60          RTS

```

```

2338 BD 51 13    POPT      LDA      LINE,X
233B BD 81 14    STA      OPT
233E EB          INX

```

```

233F BD 51 13    LDA      LINE,X
2342 99 2B 13    STA      PAR-1,Y CHANGE STORED TERMINATOR TO , OR CR 1 HOPE
2345 EB          INX

```

```

2346 10 E9      BPL      EDC
234B 4E 41 4D    NTL      .BYTE "NAME TOO LONG".CR
234B 45 20 54

```

```

234E 4F 4F 20
2351 4C 4F 4E
2354 47 9B

```

```

0023      NTLH      HILO      NTL
004B      NILL      =        NTL/256
                        -256*NTLH+NTL

```

DBPM80 - DISPLAY N BYTES
BUFFER POINTER AND LENGTH ARE ALREADY IN I0C80

```

2356 A9 0A      DBPM80 LDA      #PUTCHR
2358 BD 42 03    STA      ICCOM
235B A2 00      LDX      #0
235D 20 83 23    JBR      C101
2360 60          RTS

```

DBPLIN - DISPLAY ONE LINE OF TEXT
A=LD,X=HI ADDRESS

```

2361 20 68 23    DBPLIN JBR      DBPL1
2364 20 8E 23    JBR      BCROL
2367 60          RTS

```

DBPL1 - ADDRESS OF LINE TO I0CB

```

2368 BD 41 03    DBPL1 STA      ICBAL
236B BE 45 03    STX      ICBAL
236E A9 09      LDA      #PUTRCD+TEXT
2370 BD 42 03    STA      ICCOM
2373 A9 00      LDA      #0      SET BUFF LEN
2375 BD 49 03    STA      ICBH

```

CALLEEN DISK UTILITIES

237A 80 48 03 STA ICBLL
237D A2 00 LDX #0
237F 20 83 23 JSR CIO1
2382 60 RTS

2383 20 56 E4 CIO1 JSR CIO CALL CIO AND GO TO MENU1
2386 C0 80 CPY #80 IF BREAK KEY ABORT
2388 D0 03 BNE #+3
238A 4C 91 16 JMP MENU1
238D 60 RTS

BCROL - DO SCROLLING OF AREA BELOW MENU

238E A9 00 BCROL LDA #0
2390 AA TAX
2391 9D 49 03 STA ICBLL, X
2394 A9 0A LDA #10
2396 9D 48 03 STA ICBLL, X
2399 A9 23 LDA #ZAPH
239B 9D 45 03 STA ICBAL, X
239E A9 A7 LDA #ZAPL
23A0 9D 44 03 STA ICBAL, X
23A3 20 56 23 JSR DSPMS0
23A6 60 RTS

23A7 1C 1C 1C ZAP . BYTE CUP, CUP, CUP, CUP, CUP
23AA 1C 1C . BYTE
23AC 9C 1D 1D . BYTE DLL, CDN, CDN, CDN, CDN
23AF 1D 1D

0023 HILO ZAP
00A7 ZAPH = ZAP/256
ZAPL = -256*ZAPH+ZAP

PIDCB - POINT TOCB AT PAR(PTR)

23B1 A9 29 PIDCB LDA #PARL
23B3 18 CLC
23B4 6D AE 14 ADC PTR
23B7 9D 44 03 STA ICBAL, X
23BA A9 13 LDA #PARH
23BC 69 00 ADC #0
23BE 9D 45 03 STA ICBAL, X
23C1 60 RTS

CIOCL - CALL CIO AND PROCESS ANY ERRORS

23C2 20 56 E4 CIOCL JSR CIO CALL CIO
23C5 98 TYA
23C6 30 01 BMI #+3
23C8 60 RTS OK, RETURN
23C9 98 CIDER1 TYA ERROR STATUS
23CA 98 CIDER DEC
23CB E9 64 SBC #100
23CD A2 2F LDX #0-1 ERROR NUMS ALWAYS ARE 1XX DEC
23CF E8 CTNS INX CONVERT TENS
23D0 38 SEC
23D1 E9 0A SBC #10
23D3 10 FA RPL CTNS THE EASY (SLOW) WAY
23D5 10 CLC
23D6 69 3A LXC #10+0 CONVERT

COLLEEN DISK UTILITIES

```

23DB BE F5 23      STX      ETN
23DE A2 23          LDX      #CIEH
23E0 A9 EB          LDA      #CIEL
23E2 20 61 23      CIEH     JSR      DBPLIN
23E5 20 F8 23      JSR      CLOSX      CLOSE IOCBB 10,20
23E8 4C 91 16      JNP      MENUBL
23EB 45 52 52      CIE      .BYTE  "ERROR- 1"
23EE 4F 52 2D
23F1 20 20 20
23F4 31

```

```

23F5 00      ETN      .BYTE  0
23F6 00      EUN      .BYTE  0
23F7 9B      .BYTE  CR
                HILD    CIE
0023      CIEH      =      CIE/256
00EB      CIEL      =      -256+CIEH+CIE

```

! CLOSX - CLOSE IOCBB 10,20

```

23FB A9 0C      CLOSX     LDA      #CLOSE
23FA BD 52 03      BTB      ICCOM+10      CLOSE IOCBB 10,20
23FD BD 42 03      BTB      ICCOM+20
2400 A2 10          LDX      #10
2402 20 56 E4      JSR      CIO
2405 A2 20          LDX      #20
2407 20 56 E4      JSR      CIO
240A 60          RTS
                IF      SYSRES

```

LOAD - THIS SUBROUTINE
LOADS FROM THE FILE (MUST BE LOAD FORMAT)
INTO MEMORY, RETURNS:

X=0 LOAD OK
X=1 OPEN ERRORS Y=CIO CODE
X=2 READ ERRORS Y=CIO CODE
X=3 BAD LOAD FILE

ON ENTRY, IOCBB 1 POINTS TO FILENAME.

```

240B A2 10      LOAD      LDX      #10
240D A9 03      LDA      #OPEN
240F 9D 42 03      BTB      ICCOM, X
2412 A9 04      LDA      #4      OPEN TYPE=INPUT
2414 9D 4A 03      BTB      ICAX1, X
2417 20 56 E4      JSR      CIO      TRY TO OPEN FILE
241A 10 04      BPL      RDLF      CONT IF OK
241C A9 01      LDA      #1      OPEN ERRORS
241E D0 4F      BNE      CLFX      CLOSE AND EXIT
2420 A2 10      RDLF      LDX      #10
2422 A9 A1      LDA      #DBUFL
2424 9D 44 03      BTB      ICBAL, X
2427 A9 13      LDA      #DBUFH
2429 9D 45 03      BTB      ICBAL, X
242C A7 02      LDA      #2
242E 9D 48 03      BTB      ICBLL, X
2431 A9 00      LDA      #0
2433 9D 49 03      BTB      ICBLL, X
2436 A9 06      LDA      #GETCHR
243B 9D 42 03      BTB      ICCOM, X

```

COLLEEN DISK UTILITIES

243E 30 45	BMI	ERBT	IF ERRB
2440 A9 B4	LDA	#0B4	
2442 CD A1 13	CMF	DBUF	CHECK FOR VALID LOAD FILE
2445 D0 3A	BNE	LNL	
2447 A9 09	LDA	#009	
2449 CD A2 13	CMF	DBUF+1	
244C D0 33	BNE	LNL	
244E A2 10	RDDRC	LDX	#010
2450 A9 A1	LDA	#DBUFL	
2452 9D 44 03	STA	ICBAL, X	
2455 A9 13	LDA	#DBUFH	
2457 9D 45 03	STA	ICBAH, X	
245A A9 04	LDA	#4	
245C 9D 48 03	STA	ICBL, X	
245F A9 00	LDA	#0	
2461 9D 49 03	STA	ICBLH, X	
2464 20 56 E4	JBR	CIO	NO ERROR CHECK SO CAN CATCH EOF
2467 10 20	BPL	BTOK	IF NO ERROR
2469 C0 88	CPY	#088	SEE IF EOF
246B D0 18	BNE	ERBT	IF SOME ERROR STATUS
			IF EOF SO DONE, EXIT
246D A9 00	LDA	#0	OK STATUS
246F 4B	CLFX	PHA	
2470 98		1YA	
2471 4B		PHA	
2472 A2 10		LDX	#010
2474 A9 0C		LDA	#CLOSE
2476 9D 42 03		STA	ICCOM, X
2479 20 56 E4		JBR	CIO
247C 6B		PLA	
247D AB		TAY	
247E 6B		PLA	
247F AA		TAX	
2480 60		RTB	
2481 A9 03	LNL	LDA	#3 HAD LOAD FILE
2483 D0 EA		BNE	CLFX
2485 A9 02	ERBT	LDA	#2 READ ERRB
2487 D0 E4		BNE	CLFX
2489 A2 10	BTOK	LDX	#010
248B AD A1 13		LDA	DBUF MOVE PARAMS TO IOCB
248E 9D 44 03		STA	ICBAL, X
2491 AD A2 13		LDA	DBUF+1
2494 9D 45 03		STA	ICBAH, X
2497 AD A3 13		LDA	DBUF+2
249A 3B		BEC	
249B ED A1 13		SBC	DBUF
249E 9D 48 03		STA	ICBL, X
24A1 AD A4 13		LDA	DBUF+3
24A4 ED A2 13		BUC	DBUF+1
24A7 9D 49 03		STA	ICBLH, X
24AA FE 48 03		JNC	ICBL, X
24AD D0 03		BNE	++3
24AF FE 49 03		INC	ICBLH, X
24B2 20 56 E4		JBR	CIO READ DATA DIRECTLY TO MEMORY
24B5 30 CE		BMI	ERBT IF ERRB
24B7 4C 4E 24		JMP	RDDRC GO READ NEXT DATA REC.
			ENDIF

COLLEEN DISK UTILITIES

RETURN A=LO, X=HI. PER SET MINUS IF ERROR.
INC IPTR PAST PARAM.

24BA A9 04	GETND	LDA	#4	MAX NO DIGITS
24BC BD B0 14		STA	CTR	
24BF A9 00		LDA	#0	
24C1 BD B2 14		STA	T1	
24C4 BD B3 14		STA	T1+1	INIT TEMP TO BUILD NUMBER IN
24C7 AE AF 14	QIB	LDX	IPTR	
24CA BD 51 13		LDA	LINE,X	GET CIAR
24CD EE AF 14		INC	IPTR	
24D0 C9 9B		CMF	#CR	SEE IF TERMINATOR
24D2 F0 2B		BEG	QND	
24D4 C9 2C		CMF	#'	
24D6 F0 27		BEG	QND	
24DB 20 3A 25		JBR	HEXCON	CONVERT ASCII TO NIBBLE
24DB 30 29		BMI	ERRX	IF ERROR
24DD A0 03		LDY	#3	SHIFT T1,T1+1 BY 4
24DF 1B	BHT1	CLC		
24E0 2E B3 14		RCL	T1+1	
24E3 2E B2 14		RCL	T1	
24E6 BB		DEY		
24E7 10 F6		BPL	BHT1	
24E9 0D B3 14		ORA	T1+1	OR IN NEW NIBBLE
24EC BD B3 14		STA	T1+1	
24EF CE B0 14		DEC	CTR	COUNT DIGIT
24F2 10 D3		BPL	QIB	LOOP UNLESS TOO MANY DIGITS
24F4 A9 0C		LDA	#TMDL	
24F6 A2 25		LDX	#TMDH	
24F8 20 61 23	ERRX1	JBR	DBPLIN	
24FB CE A2 14		DEC	PER	
24FE 60		RTS		
24FF AD B3 14	QND	LDA	T1+1	
2502 AE B2 14		LDX	T1	
2505 60		RTS		
2506 A9 1C	ERRX	LDA	#IHPL	INVALID HEX PARAM
2508 A2 25		LDX	#IHPH	
250A D0 EC		BNE	ERRX1	
250C 54 4F 4F	TMD	.BYTE	"TOO MANY DIGITS".CR	
250F 20 4D 41				
2512 4E 59 20				
2515 44 49 47				
2518 49 54 53				
251B 9B				

0025	TMDH	=	TMD/256
000C	TMDL	=	-256*TMDH+TMD
251C 49 4E 56	IHP	.BYTE	"INVALID HEXADECIMAL PARAMETER".CR

251F 41 4C 49
2522 44 20 4B
2525 45 5B 41
2528 44 45 43
252B 49 4D 41
252E 4C 20 50
2531 41 52 41
2534 4D 45 54
2537 45 52 9B

0025	IHPH	=	IHP/256
------	------	---	---------

COLLEEN DIBK UTILITIES

HEXCON - CONVERT ASCII CHAR IN A TO HEX NIBBLE IN A. RETURN
MINUS CONDITION, A=FF IF ERROR.

253A 38	HEXCON	BEC	
253B E9 30		BBC	#0
253D 30 0F		BMI	ERRX2 ASCII BELOW '0'
253F C9 0A		CMP	#10
2541 30 0D		BMI	OKX 0-9 CONVERTED SO EXIT
2543 38		BEC	
2544 E9 07		BBC	#A-'0-10
2546 C9 0A		CMP	#10 CONVERTED VALUE MUST BE 10 OR MORE
254B 30 04		BMI	ERRX2 BETWEEN '9' AND 'A'
254A C9 10		CMP	#10
254C 30 02		BMI	OKX A-F CONVERTED
254E A9 FF	ERRX2	LDA	#FF
2550 C9 00	OKX	CMP	#0 SET STATUS BY VALUE IN A
2552 60		RTB	

COLLEEN DISK UTILITIES

OETDN - GET A DEVICE NUMBER FROM LINE(IPTR)
RETURN IT IN A

2553 2C A2 14 OETDN BIT PER SEE IF PARAM ERROR ALREADY
2556 30 27 BMI QDR IF SO DONT BOTHER

2558 AE AF 14 LDX IPTR
255B BD 51 13 OETD LDA LINE,X
255E EB INX

255F C9 44 CMP #D IF DN
2561 F0 FB BEQ OETD OO GET DIOIT
2563 3B SEC

2564 E9 30 BBC #0 CONVERT DIOIT
2566 F0 1B BEQ BDB CANT BE ZERO
2568 30 16 BMI BDB IF NOT DIOIT

256A C9 05 CMP #5
256C 10 12 BPL BDB TOO LARGE
256E 4B PHA

256F BD 51 13 ODI LDA LINE,X
2572 EB INX

2573 C9 2C CMP #1
2575 F0 04 BEQ ODX IF TERMINATOR

2577 C9 9B CMP #CR
2579 D0 F4 BNE ODI KEEP LOOKING
257B BE AF 14 ODX BTX IPTR ADVANCE POINTER

257E 6B PLA
257F 6B ODH RTB

2580 CE A2 14 BHS DEC PER
2583 A9 8B LDA #NDBL NEED DEVICE SPEC MSG
2585 A2 25 LDX #NDBH

2587 20 61 23 JBR DBPLIN
258A 6B RTB

258B 4E 45 45 NDB , BYTE "NEED D1 THRU D4", CR

258E 44 20 44

2591 31 20 54

2594 4B 32 55

2597 20 44 34

259A 9B

HIL0 NDB
NDBH = NDB/256
NDBL = -256*NDBH+NDB

DRH - DEVICE REDIRECTION HANDLER

WHEN LOGICAL DEVICES ARE CREATED, THE HANDLER ADDRESS
POINTS HERE. THE DRH LOOKS UP THE PHYSICAL DEVICE SPEC
AT OPEN TIME AND CALLS THE APPROPRIATE HANDLER. THE IOCB
HANDLER ID IS THEN SET TO THAT HANDLER SO CIO CALLS AFTER
THE OPEN GO DIRECTLY TO THAT HANDLER.

259B DRH =
HIL0 DRH
0025 DRH/256
009B DRH = -256*DRH+DRH

HANDLER VECTORS (ONLY OPEN IS USED)

9B F7 25
259D 00 00 00 LINT
HARD EOPEN-1
BYTE 0,0,0,0,0

COLLEEN DISK UTILITIES

25A7 60	***S RTB	UNUSED INIT ROUTINE
25AB	PDBT ***B0	TABLE OF PHYSICAL DEV:FILESPEC
0025	PDBTH HILO	PDBT
00AB	PDBTL =	PDBT/256
		-256*PDBTH+PDBT
25FB A0 00	EDPEN LDY	NO
25FA B1 24	LDA	(ICBALZ),Y GET LOGICAL DEVICE NAME
25FC 20 2E 1C	JBR	FLDS SEARCH LOG DEV NAME TABLE
25FF 10 03	BPL	FLD IF FOUND IT OK
2601 A0 AC	INTERNAL ERROR IF WE GOT HERE.	
2603 60	IERR LDY	***AC NEW ERROR
2604 BA	HERRX RTB	
2605 A6 2E	FLD TXA	CREATE POINTER TO PHYSICAL DEVSPEC
2607 0A	LDX	ICIDNO GET IOCB #
2608 0A	ASL	A
2609 0A	ASL	A
260A 0A	ASL	A
260B 69 AB	ADC	#PDBTL
260D 85 24	STA	ICBALZ
260F 9D 44 03	STA	ICBAL,X FOR FMS
2612 A9 00	LDA	NO
2614 69 25	ADC	#PDBTH
2616 85 25	STA	ICBAHZ
2618 9D 45 03	STA	ICBAH,X FOR FMS
261B 20 9E E6	JBR	DEVSRG GO FIND PHYS DEVICE HID/DNO
261E 80 E3	BCB	HERRX IF ERRORS
2620 A5 21	LDA	ICDNOZ
2622 9D 41 03	STA	ICDNO,X FOR FMS
2625 20 3D E6	JBR	COMENT GET PHYS DEV HOLR OPEN ENTRY
2628 80 D9	BCB	HERRX
262A 20 89 E6	JBR	GOHAND GO DO REAL OPEN
262D 60	RTB	THATS ALL
	END	

CULLFEN DISK UTILITIES

SYMBOL TABLE			
AAM	1FB7	ADDC	22EB
ANL	001D	ASP	1FA4
HDS	2580	BLF	1CE5
BRKKEY	0011	BRMO	1D3E
DK2	1BD1	BSM	1BD7
CARTST	8FFA	CBIT	1FA0
CDSK	1AF2	CEND	1E06
CFTE2	2316	CHRO1	223B
CHROET	2233	CIE	23EB
CIFX	23E2	CIO	E456
CIOER	23CA	CIOER1	23C9
CKRB	1A27	CLD1	1FDA
CLFX	246F	CLOC	1A33
CLOSE	000C	CLOBX	23FB
CPMG	190D	CPBL	1C12
CR	009B	CRD	1C0A
CRRCH	0014	CBRCL	00AB
CUP	001C	DB1	1421
DR3	139E	DB3H	0013
DBLL	0000	DBUF	13A1
DBUFL	00A1	DBUFLO	0304
DD1	1BA0	DDMO	1D5A
DEDE	1855	DEFDEV	1B76
DELX	18D5	DENO	18FC
DFCX	2072	DTRLBT	172D
IN L	009C	DLMO	179B
DMEND	1607	DMENU	14C7
DDCPY	19DA	DORD	1E9C
DOSOSH	0016	DOSOSL	0025
DOW	1EBE	DPFM	2009
DRIL	009B	DRTS	1FEC
DSHI	030B	DSHIH	0003
DSH	00A7	DSLO	030A
DSPL1	236B	DSPLIN	2361
DSX	1BC0	DTH	14BB
DU1	1650	DU3	1A51
DUJPTH	0016	DUJPTL	0007
DULENL	0040	DUNIT	0301
DUPFIL	201F	DWQ	17F7
ECL	00EE	EDH	0014
ENUF	1E66	EDC	2331
ERRX1	24F8	ERRX2	254E
EUN	23F6	FDP	1AF1
FLD	2604	FLDB	1C2E
FMINIT	078B	FMS	0700
FORMAT	00FE	QC1	19DC
QDX	257B	QETCHR	0006
GETFIL	229E	GETIC1	2284
GETNO	248A	GETRCD	0004
QND	24FF	QOHAND	E689
HATABS	031A	HATB	1C21
HERRX	2603	HEXCON	253A
IBDL	000E	ICAX1	034A
ICBAIZ	0025	ICBAL	0344
ICBL1	034B	ICCON	0342
ICI	0340	ICHIDZ	0020
IDD	1F63	IDDH	001F
AF	131D	ASPT	1ED2
BLFH	001C	BRUN	1B2C
BSM1	001B	CBES	14AC
CITE	22F4	CFTE	22F4
CIRQ2	224F	CIRQ3	2260
CIEH	0023	CIFL	00EB
CID1	23B3	CIOCL	23C2
CIOOET	220D	CKMEM	1FF2
CLNKH	1FD2	CLF	001E
CLOOP	19FA	CLOB1	18DB
CLSCR	007D	COMENT	E63D
CPTR1	200B	CPYFIL	1936
CRT	001F	CSRC	14AB
CTNS	23CF	CTR	14B0
DB1H	0014	DRIL	0021
DB3L	009E	DBLH	0001
DBUFH	0013	DBUFH1	0305
DCB	0300	DCOMND	0302
DDSK	1AEF	DDXT	1FED
DELETE	0021	DELFIL	17BA
DEVSRG	E69E	DF1	17E3
DKIND	E453	DKWRT	1FD0
DLSTO	178B	DLST1	178B
DMENUH	0014	DMENUL	00C7
DOS	1300	DOSOS	1625
DOSVEC	000A	DOSHDP	1F0Q
DRH	259B	DRIM	0025
DS	1CA7	DSH	001C
DSHIL	000B	DSKUTL	1650
DSLOH	0003	DSLCL	000A
DSPMBO	2356	DSTATB	0303
DTH1	0014	DTIL	00BB
DU4	1A44	DUJPT	1607
DULEN	0140	DULEN1	0001
DUNUM	000F	DUPDSK	1DCQ
EC	21EE	ECH	0021
EDL	00C1	EDN	14C1
EDPEN	25FB	ERRX	2506
ERBT	24B5	ETN	23F5
FDPH	001A	FNPL	00F1
FLDX	1C3B	FL1	1C30
FMTDSK	1AB7	FMX	1ABD
GN1	256F	GNR	257F
GETD	255B	GETDN	2553
GETLIN	21F1	GETNAM	2290
GNB	24C7	GNF	1759
GT1	22DD	GT2	22D5
HATSL	1C23	HATX	1C2D
IBD	1E0E	IBDH	001E
ICAX2	034B	ICBAH	0345
ICBALZ	0024	ICBLH	0349
ICDNO	0341	ICDNOZ	0021
ICIDNO	002E	ICSTA	0343
IDD1	0063	INRD	1B2A

COLLEEN DISK UTILITIES

IOD	1EC1	IPTR	14AF	IPTRH	0014	IPTRL	00AF
IRGEN	D20E	IBCAR	1806	ISD	1F44	ISDH	001F
ISHL	0044	JMPTBL	001B	KDN	14C4	KDNH	0014
KIML	00C4	LBUFH	0013	LBUFL	0051	LDLIL	1CB1
LDFX	1CD5	LDND	14BD	LDNT	259D	LDST	14BB
LFMO	1CF3	LINE	1351	LKFL	1D0B	LKMO	1D1D
LNL	24B1	LOAD	240B	LOADV	1319	LOCK	0023
LNS	1EAE	LRB1	1FAB	MAXDEV	0021	MDN1	187C
MIN2	18BC	MDN3	1874	MENLO	02E7	MENTOP	02E5
MENUBL	1691	MENUBZ	14A1	MNA	14B6	MNL	14BB
MNGL	1691	MNSLH	0016	MNSLL	0091	NCA	1B1E
NCAH	001B	NCAL	001E	NDF	17D3	NIN-H	0017
NDF-L	00D3	NDS	25BB	NDSH	0025	NDSL	00BB
NF	1CDB	NORM	1E5C	NRH	1F34	NRMH	001F
NIML	0034	NSEC	1DF5	NBI	16E2	NSIH	0016
NSIL	00E2	NTL	234B	NTLH	0023	NTLL	004B
NXIOCB	21CD	NXB	1FB2	ODMS	19AD	OF	191D
OFH	0019	DEL	001D	OK	1D77	OKH	001D
OKL	0077	OKB	207F	OKX	2550	OL	2264
OLH	0022	OLL	0064	OPDES	19B7	OPDES1	19C3
OPDES3	19C5	OPEN	0003	OPSRC	1971	OPSRC1	19B2
PAR	14B1	PAR	1329	PARH	0013	PARL	0029
PIEB	199B	PDEB1	19A2	PDST	25A8	PDSTH	0025
PDSTL	00AB	PER	14A2	PERX	2279	PERX1	227F
PIDCB	23D1	PMB	171B	PMBH	0017	PMBL	001B
POKMSK	0010	POPT	233B	PBMEN	16FB	PSRC	1959
PTR	14AE	PTRH	0014	PTHL	00AE	PUTCHR	000A
PUTRCD	000B	QWMO	1CBB	RAMLO	001A	RANGE	16DB
RCNT	14A4	RDDRC	244E	RDFN	1857	RDLF	2420
RDM	14BA	RENAME	0020	RENEIL	1A54	RNM	1A6C
RSEC	1FBB	RSEC1	1FC5	RUNAD	02E0	RVTOC	1D9A
SAVFIL	20DB	SAVH	14BF	SAVHH	0014	SAVHL	00BF
SAVX	14AD	SCMO	1B2B	SCRDL	239E	SND	1E2D
SE1VBV	E45C	BFMO	21A0	BHFLM	02BE	BHMEN	1671
SH11	24DF	BIT	16EF	BITH	0016	BITL	00EF
SSTAT	14A5	STCAR	1AF5	STDD	1EE9	STDD1	1EF8
STDD2	1EEE	BTE	2325	STF	18BC	STOK	24B9
STVEC	14B4	SUI	1BF5	SU2	1BEF	SWATH	1F2F
SWATL	1F2A	SWDP	14A6	SWLOP	1F02	SYBRES	0001
SYBVBV	E45F	SYVBL	E45F	SYVBLH	00E4	SYVBLL	005F
TI	14B2	TEXT	0001	TF	18C6	TFH	001B
TFL	00C6	TMD	250C	TMDH	0025	TMH	000C
TYG	1BE6	TYGH	001B	TYGL	00E6	ULFIL	1D30
ULMO	1D45	UNLOCK	0024	UNNO	14A3	USRDOS	1700
VFM	1AD7	VFPH	001A	VERL	00D7	WARNST	000B
WMO	1C6F	WBMCH	001C	WBMCL	006F	WROOT	1C39
WPH	1C6C	WDR	214E	WHD	1AC0	WIMH	001A
WIDL	00C0	XBLK	1EA5	XITVBV	E462	XIVBL	E462
XIVBLH	00E4	XTVBL	0062	ZAP	23A7	ZAPH	0023
ZAPL	00A7						