



CONSULTING ENGINEERS
COMPUTER APPLICATIONS

SORRENTO VALLEY ASSOCIATES, INC.

APP-L-CACHE

EPROM Source Listing

REV D

14JUL82

PAGE - 0

Current memory available: 8145

```
00001 ;-----  
00001 ;  
00001 ;  
00001 ; APP-L-CACHE ROM SOURCE  
00001 ;  
00001 ; Copyright (c) 1982  
00001 ; Sorrento Valley Assoc Inc.  
00001 ;  
00001 ; ALLOWS RAM DISC TO RUN  
00001 ; UNDER DOS 3.3 AND  
00001 ; AND PASCAL VERSION 1.1  
00001 ;  
00001 ; HELD IN 2716 ON BOARD  
00001 ;  
00001 ;-----  
00001 ;  
00001 ;  
00001 ;  
00001 ; CHANGES: REV D - J.C. 22  
00001 ;  
00001 ; ALLOWS USE WITH  
00001 ; ROUTINE ADDED 1  
00001 ; INTERCEPT IN PA  
00001 ;  
00001 ;  
00001 ;  
00001 ;  
00001 ; ABSOLUTE
```

2 blocks for procedure code 7669 words left

```

0000:          .PROC CONTROL
Current memory available: 7620
0000:          .ORG 0C800
C800:
C800:
C800:          ; **** SCRATCH FOR DOS READ/WRITE ****
C800: 0006      VOLNO  .EQU 6    ;VOLUME NUMBER
C800: 0007      DRVNO  .EQU 7    ;DRIVE NO
C800: 0008      TRNO   .EQU 8    ;TRACK NUMBER
C800: 0009      SCNO   .EQU 9    ;SECTOR NUMBER
C800: 0048      IOB    .EQU 48   ;I/O TABLE
C800:
C800:          ; **** SCRATCH FOR DOS + PASCAL ****
C800: 0019      PGAD   .EQU 19   ;PAGE ADDRESS
C800: 001B      BFAD   .EQU 1B   ;DOS BUFFER ADDRESS
C800: 001D      BNKNO  .EQU 1D   ;BANK NUMBER
C800: 001E      WRTEN  .EQU 1E   ;STATUS OF BANKS
C800: 001F      BNKSTAT .EQU 1F   ; COMMON TO DOS AND PASCAL
C800: 0019      TEMP1   .EQU PGAD ;TEMPORARY
C800: 001B      TEMP2   .EQU BFAD ; *
C800:
C800:          ; **** DOS EQUATES ****
C800: BCFE      .EQU 0BCFE ;SIZE OF RAMDISC
C800: BCFF      .EQU 0BCFF ;ERRORS NOT FLAGGED IF THIS LOC HOLDS 55H
C800: 0000      K256   .EQU 0    ;FLAG FOR 256K BOARD
C800: 0001      K192   .EQU 1    ; * * 192K *
C800: 0002      K128   .EQU 2    ; * * 128K *
C800: 0003      K64    .EQU 3    ; * * 64K *
C800: 0037      VLNO   .EQU 55.  ;VOL NO OF DISC
C800:
C800:          ;ENTRY POINTS
C800:
C800: 4C 0000      JMP 0
C803:
C803: 4C ****      DOS33   JMP DDISK
C806:
C806: 4C ****      PASCAL  JMP PDISK
C809:
C809: 4C ****      RSTRBNK JMP RESTOREBNK
C80C:
C80C: 4C ****      SAVERBK JMP SAVERBKSTAT
C80F:
C80F: 4C ****      RAMSIZE JMP SAFERSIZE
C812:
C812: 4C ****      PASBOOT JMP AXBNKINIT
C815:
C815: ****        DRLIMITS .WORD LINTAB      ;address of limit table for dos drive
C817:
C817:
C804* 17C8
C817: 84 48      DDISK   STY IOB
C819: 85 49      STA IOB+1
C81B: A9 10      LDA #10
C81D: A0 01      LDY #1
C81F: D1 48      CMP #IOB,Y

```

```

C821: F0**           BEQ DRAM ;IF NOT SLOT 1
C823: A4 48          LDY IOB
C825: A5 49          LDA IOB+1
C827:
C827: 4C 00BD        JMP 0BDOO ; JMP BACK TO RWTS
C82A:
C821* 07
C82A: 20 ****        DRAM   JSR SAVEBKSTAT
C82D: 2C 8BC0          BIT 0C08B
C830: 2C 8BC0          BIT 0C08B
C833: A0 02          LDY #2
C835: B1 48          LDA @IOB,Y
C837: 85 07          STA DRVNO ;GET DRIVE NO
C839: C8
C83A: C8
C83B: B1 48          LDA @IOB,Y
C83D: 85 08          STA TRNO ;GET TRACK NO
C83F: C8
C840: B1 48          LDA @IOB,Y
C842: 85 09          STA SCNO ;GET SECTOR NO
C844: A9 00          LDA #0
C846: 85 19          STA TEMP1
C848: C6 07          DEC DRVNO
C84A: F0**          BEQ $1    ;BIF DRIVE 1
C84C: AD ****        LDA LIMTAB+1
C84F: 85 19          STA TEMP1
C851: A9 08          LDA #8
C84A* 07
C853: 85 1B          $1    STA TEMP2
C855: AD FEBC        LDA SZFLG
C858: 0A              ASL A
C859: 18              CLC
C85A: 65 1B          ADC TEMP2
C85C: AA              TAX      ;index into limit table is DRIVE*8+SZFLG*2
C85D: A5 08          LDA TRNO
C85F: 38              SEC
C860: FD ****        SBC LIMTAB,X ;1st entry in limit table is start track
C863: 85 08          STA TRNO
C865: E8              INX
C866: DD ****        CMP LIMTAB,X ;2nd entry is no of available tracks
C869: 90**          BCC CONT
C86B: 4C ****        JMP DRERROR
C86E:
C86E:                 ;LIMIT TABLE defines start track and no of tracks of different size systems.
C86E:                 ;
C867* 6EC8
C861* 6EC8
C84D* 6FC8
C815* 6EC8
C86E: 00 23          LIMTAB .BYTE 0,23 ;256K
C870: 00 23          .BYTE 0,23 ;192K      DRIVE 1
C872: 07 1C          .BYTE 7,1C ;128K
C874: 11 0C          .BYTE 11,0C ;64K
C876:
C876: 0A 19          .BYTE 0A,19 ;256K

```

C878I 11 09		.BYTE 11,09	;192K	DRIVE 2
C87A1 00 00		.BYTE 0,0	;128K	
C87C1 00 00		.BYTE 0,0	;64K	
C87E1				
C869* 13				
C87E1	CONT:			
C87E1 A0 0E		LDY #0E		
C880I A9 37		LDA #ULMO		
C882I 91 48		STA @IOB,Y	;SET VOLUME NO	
C884I C8		IMY		
C885I A9 10		LDA #10		
C887I 91 48		STA @IOB,Y	;SET PREVIOUS SLOT	
C889I C8		IMY		
C88AI E6 07		INC DRVNO		
C88CI A5 07		LDA DRVNO		
C88EI 91 48		STA @IOB,Y	;SET PREVIOUS DRIVE NO	
C890I A5 19	\$2	LDA TEMP1		
C892I 18		CLC		
C893I 65 08		ADC TRMO	;BNKNO=BNKNO+TRMO	
C895I 85 1D		STA BNKNO		
C897I 30**		BMI DRERROR		
C899I C9 3C		CMP #3C		
C89BI B0**		BCS DRERROR	;IF BNKNO<0 OR BNKNO>3B THEN DRIVE ERROR	
C89DI A4 07		LDY DRVNO		
C89FI C9 23		CMP #23		
C8A1I 90**		BCC \$3	;BIF DRIVE SHOULD BE 1	
C8A3I C0 02		CPY #2		
C8A5I D0**		BNE DRERROR	;BIF DRIVE NOT 2	
C8A7I F0**		BEQ \$4		
C8A1* 06				
C8A9I C0 01	\$3	CPY #1		
C8AB1 D0**		BNE DRERROR	;BIF DRIVE NOT 1	
C8A7* 04				
C8AD1 A9 3B	\$4	LDA #3B		
C8AF1 38		SEC		
C8B0I E5 1D		SBC BNKNO		
C8B2I 85 1D		STA BNKNO	;BNKNO=3B-BNKNO	
C8B4I A5 09		LDA SCNO		
C8B6I 09 D0		ORA #0D0		
C8B8I 85 1A		STA PGAD+1		
C8BAI A9 00		LDA #0		
C8BCI 85 19		STA PGAD	;PGAD=0D000+SCNO*256	
C8BEI				
C8BEI A0 0C		LDY #0C		
C8C0I B1 48		LDA @IOB,Y	;TEST REQUIRED OPERATION	
C8C2I F0**		BEQ NOERROR	;IF NULL RETURN	
C8C4I				
C8C4I C9 04		CMP #4		
C8C6I F0**		BEQ NOERROR	;IF FORMAT RETURN	
C8C8I				
C8C8I A5 1D	RW	LDA BNKNO		
C8CAI 8D 8FC0		STA 0C08F	;SELECT BANK	
C8CDI A0 08		LDY #8		
C8CFI B1 48		LDA @IOB,Y		
C8D1I 85 18		STA BPAD		

```

C8D3: C8      INY
C8D4: B1 48    LDA @I0B,Y ;GET ADDRESS OF DOS
C8D6: 85 1C    STA BFAD+1 ;   BUFFER
C8D8: A0 0C    LDY #0C
C8DA: B1 48    LDA @I0B,Y
C8DC: C9 01    CMP #1
C8DE: D0**    BNE WRT
C8E0: AD 8BC0  LDA OC08B
C8E3: A0 00    LDY #0
C8E5: B1 19    LOAD   LDA @PGAD,Y
C8E7: 91 1B    STA @BFAD,Y ;MOVE SECTOR FROM
C8E9: C8      IMY     ;DISKRAM TO DOS
C8EA: D0F9    BNE LOAD ;   BUFFER
C8EC: F0**    BEQ NOERROR
C8EE:
C8DE* 0E
C8EE: AD 8BC0  WRT    LDA OC08B
C8F1: AD 8BC0  LDA OC08B
C8F4: A0 00    LDY #0
C8F6: B1 1B    STORE   LDA @BFAD,Y
C8F8: 91 19    STA @PGAD,Y ;MOVE SECTOR FROM
C8FA: C8      IMY     ;DOS BUFFER TO
C8FB: D0F9    BNE STORE ; DISKRAM
C8EC* 0F
C8C6* 35
C8C2* 39
C8FD: 20 ****  NOERROR JSR RESTOREBNK
C900: 18      CLC
C901: 60      RTS      ;EXIT TO DOS
C902:
C8AB* 55
C8A5* 58
C89B* 65
C897* 69
C86C* 02C9
C902: AD FFBC  DRERROR LDA REFLG
C905: C9 55    CMP #55
C907: F0F4    BEQ NOERROR ;IF REFLG HOLDS 55H THEN DO NOT FLAG ERROR
C909: A9 40    LDA #40
C90B: A0 0D    ERROR   LDY #0D
C90D: 91 48    STA @I0B,Y ;SET ERROR CODE
C90F: 20 ****  JSR RESTOREBNK
C912: 38      SEC
C913: 60      RTS      ;FLAG ERROR RETURN TO DOS
C914:
C914:          ;*****
C914:          ; ROUTINE COMMON TO DOS AND PASCAL
C914:
C82B* 14C9
C80D* 14C9
C914:          SAVEBNKSTAT:
C914: AD 8FC0    LDA OC08F
C917: 85 1F    STA BNKSTAT ;SAVE NO. OF CURRENTLY SELECTED BANK
C919: A9 00    LDA #0
C91B: 85 1E    STA WRTEM

```

```

C91D: 2C 83C0          BIT 0C083
C920: AD FFFF          LDA 0FFFF
C923: EE FFFF          INC 0FFFF      ;TEST IF RAM IS WRITE PROTECTED
C926: CD FFFF          CMP 0FFFF
C929: D0**              BNE $1
C92B: E6 1E              INC WRTEM    ;SAVE RESULT OF TEST
C929* 02
C92D: CE FFFF          $1 DEC 0FFFF
C930: 60                  RTS
C931:
C931:
C910*: 31C9
C8FE*: 31C9
C80A*: 31C9
C931:
C931: 2C 82C0          RESTOREBNK:
C934: A5 1E              BIT 0C082
C936: D0**              LDA WRTEM
C938: 2C 83C0          BNE $1      ;RESTORE WRITE ENABLE/PROTECTED
C93B: 2C 83C0          BIT 0C083
C936*: 06
C93E: A5 1F              $1 LDA BNKSTAT
C940: 8D 8FC0          STA 0C08F
C943: 29 C0              AND #0C0      ;RESTORE
C945: 30**              BMI ENROM    ;DISKRAM STATUS
C947: D0**              BNE ENAUX
C949: AD 83C0          LDA 0C083
C94C: 4C ****          JMP FIN
C947*: 06
C94F: AD 88C0          ENAUX LDA 0C08B
C952: 4C ****          JMP FIN
C945*: 0E
C955: 29 40              ENROM AND #40
C957: D0**              BNE ENRX
C959: AD 81C0          LDA 0C081
C95C: 4C ****          JMP FIN
C957*: 06
C95F: AD 89C0          ENRX LDA 0C089
C95D*: 62C9
C953*: 62C9
C94D*: 62C9
C962: 60                  FIN RTS
C963:
C963:
C810*: 63C9
C963: SAFERSIZE:
C963: 20 14C9          JSR SAVEBNKSTAT
C966: 20 ****          JSR RSIZE
C969: 20 31C9          JSR RESTOREBNK
C96C: 60                  RTS
C96D:
C96D:
C967*: 6DC9
C96D: 2C 8BC0          RSIZE BIT 0C08B
C970: 2C 8BC0          BIT 0C08B

```

C973: A9 00		LDA #0	
C975: A8		TAY	;Y - size
C976: AA		TAX	;X - current bank
C977: 8E 8FC0	SETUP	STX 0C08F	;select bank
C97A: AD 00D0		LDA 0D000	;save contents of test location
C97D: 48		PHA	
C97E: A9 55		LDA #55	
C980: 8D 00D0		STA 0D000	
C983: 20 ****		JSR DELAY	;delay to allow A/C data lines to settle down
C986: CD 00D0		CMP 0D000	
C989: D0**		BNE NXT1	;bif no ram present
C98B: A9 30		LDA #30	
C98D: 8D 8FC0		STA 0C08F	
C990: AD 00D0		LDA 0D000	
C993: C9 55		CMP #55	;test if ram common with bank 30 ,
C995: 08		PHP	;save res on stack
C996: 8E 8FC0		STX 0C08F	;restore current bank
C999: A9 AA		LDA #0AA	
C99B: 8D 00D0		STA 0D000	
C99E: 20 ****		JSR DELAY	;delay to allow A/C data lines to settle down
C9A1: CD 00D0		CMP 0D000	
C9A4: D0**		BNE NXTBNK	;bif no ram present
C9A6: A9 30		LDA #30	
C9A8: 8D 8FC0		STA 0C08F	;test if ram common with bank 30
C9AB: AD 00D0		LDA 0D000	
C9AE: C9 AA		CMP #0AA	
C9B0: D0**		BNE DONE	;bif ram present and not common with bank 30
C9B2: 28		PLP	
C9B3: D0**		BNE DON1	; " " " " " " " " "
C9B5: E0 30		CPX #30	
C9B7: F0**		BEQ DON1	;bif ram present in bank 30
C9B9: D0**		BNE NXT1	
C9A4* 15			
C9BB: 68	NXTBNK	PLA	
C9B9* 01			
C9B9* 31			
C9BC: C8	NXT1	IMY	;inc size flag
C9BD: 8A		TXA	
C9BE: 18		CLC	
C9BF: 69 10		ADC #10	
C9C1: AA		TAX	;X - next bank
C9C2: 68		PLA	
C9C3: C0 04		CPY #4	
C9C5: 90B0		BCC SETUP	;bif next bank
C9C7: 60		RTS	
C9B0* 16	DONE	PLA	
C9C8: 68			
C9B7* 10			
C9B3* 14			
C9C9: 68	DON1	PLA	
C9CA: 8D 00D0		STA 0D000	;restore test location
C9CD: 60		RTS	
C99F* CEC9	DELAY	PHA	
C984* CEC9			
C9CE: 48			

```

C9CF: 98          TYA
C9D0: 48          PHA
C9D1: A0 00       LDY #0
C9D3: C8          $1   INY
C9D4: D0FD       BNE $1
C9D6: 68          PLA
C9D7: A8          TAY
C9D8: 68          PLA
C9D9: 60          RTS
C9DA:
C9DA: ;*****
C9DA: ;      USED IN PASCAL SYSTEM
C9DA:
C9DA: VERSION ,EQU 0BF21 ;ADDRESS OF PASCAL VERSION NO.
C9DA:
C9DA: ;**** SCRATCH FOR PASCAL ****
C9DA: 001B ,EQU BFAD ;TRANSFER ADDRESS
C9DA: 0020 ,EQU 20  ;START BLOCK NUMBER
C9DA: 0022 ,EQU 22  ;READ=1 WRITE=0
C9DA: 002A ,EQU 2A  ;BANK NUMBER
C9DA: 0026 ,EQU 26  ;NUMBER OF BYTES TO BE TRANSFERRED
C9DA:
C807* DAC9
C9DA: PDISK:
C9DA: 68          PLA
C9DB: A8          TAY
C9DC: 68          PLA
C9DD: AA          TAX    ;SAVE RETURN ADDRESS IN X,Y
C9DE:
C9DE: 68          PLA
C9DF: 85 20       STA BLKNUM
C9E1: 68          PLA
C9E2: 85 21       STA BLKNUM+1 ;GET STARTING BLOCK
C9E4:
C9E4: 68          PLA
C9E5: 85 26       STA BYTESTOTRAN
C9E7: 68          PLA
C9E8: 85 27       STA BYTESTOTRAN+1 ;GET # OF BYTES TRANSFER
C9EA:
C9EA: 68          PLA
C9EB: 85 1B       STA TRANADDR
C9ED: 68          PLA
C9EE: 85 1C       STA TRANADDR+1 ;GET TRANSFER ADDRESS
C9F0:
C9F0: 68          PLA    ;DROP UNIT NUMBER
C9F1: 68          PLA    ;ALREADY OBTAINED
C9F2:
C9F2: AD 21BF     LDA VERSION
C9F5: C9 00       CMP #0    ;IF PASCAL VERSION 1.1
C9F7: F0**       BEQ $1    ; THEN
C9F9: 68          PLA    ;DROP CONTROL
C9FA: 68          PLA    ; WORD
C9FB:
C9F7* 02
C9FB: 8A          $1   TXA

```

```

C9FC: 48          PHA
C9FD: 98          TYA
C9FE: 48          PHA      ;RESTORE RETURN ADDRESS
C9FF:
C9FF:
C9FF:
C9FF: A5 20        BLKTRN LDA BLKNUM
CA01: 38          SEC      ;ADJUST BLOCK NUMBER TO
CA02: E9 02        SBC #2   ;TO ENABLE USE OF BLOCKS 0 AND 1
CA04: 85 2A        STA BNKNUM
CA06: 85 20        STA BLKNUM
CA08: A5 21        LDA BLKNUM+1
CA0A: E9 00        SBC #0
CA0C: B0**        BCS $1    ;IF ATTEMPT TO ADDRESS
CA0E: A2 03        LDX #3    ;BLOCKS 0 OR 1
CA10: 60          RTS      ; THEN REPORT BAD MODE ERROR
CA0C* 03
CA11: 85 2B        $1      STA BNKNUM+1
CA13: 85 21        STA BLKNUM+1
CA15:
CA15:             ;CALC BANK NUMBER
CA15:
CA15: 46 2B        LSR BNKNUM+1
CA17: 66 2A        ROR BNKNUM
CA19: 46 2A        LSR BNKNUM
CA1B: 46 2A        LSR BNKNUM
CA1D: A9 FF        LDA #OFF
CA1F: 45 2A        EOR BNKNUM
CA21: 18          CLC
CA22: 69 3C        ADC #3C
CA24: 85 2A        STA BNKNUM
CA26:
CA26:             ;CALC START ADDRESS
CA26:
CA26: A5 20        LDA BLKNUM
CA28: 29 07        AND #07
CA2A: 0A          ASL A
CA2B: 69 D0        ADC #0DO
CA2D: 85 1A        STA PGAD+1
CA2F: A9 00        LDA #00
CA31: 85 19        STA PGAD
CA33:
CA33: 20 14C9      JSR SAVEBNKSTAT ;SAVE BANK STATUS
CA36:
CA36: A5 2A        LDA BNKNUM
CA38: 8D 8FC0      STA OC08F   ;SELECT BANK NO
CA3B:
CA3B:
CA3B: A5 22        LDA RWFLG
CA3D: F0**        BEQ PWRITE
CA3F:
CA3F:             ;LOAD DATA FROM DISK RAM TO MEMORY
CA3F: 2C 8BC0      BIT OC08B
CA42: A6 27        LDX BYTESTOTRAM+1
CA44: F0**        BEQ PRTRD

```

CA46: A0 00	PLOAD	LDY #0
CA48: B1 19	\$1	LDA @PGAD,Y ;TRANSFER FULL
CA4A: 91 1B		STA @TRANADDR,Y ; 256 BYTE BLOCKS
CA4C: C8		INY
CA4D: D0F9		BNE \$1
CA4F: E6 1C		INC TRANADDR+1
CA51: E6 1A		INC PGAD+1
CA53: A9 E0		LDA #0E0
CA55: C5 1A		CMP PGAD+1
CA57: F0**		BEQ NPCLD
CA59: CA	CNTLD	DEX
CA5A: D0EA		BNE PLOAD
CA44*: 16		
CA5C: A6 26	PRTRD	LDX BYTESTOTRAN
CA5E: F0**		BEQ COMPLETE
CA60: A0 00		LDY #0
CA62: B1 19	\$1	LDA @PGAD,Y ;TRANSFER PARTIAL
CA64: 91 1B		STA @TRANADDR,Y ; 256 BYTE BLOCK
CA66: C8		INY
CA67: CA		DEX
CA68: D0F8		BNE \$1
CA6A: F0**		BEQ COMPLETE
CA57*: 13		
CA6C: C6 2A	NPCLD	DEC BNKNUM
CA6E: A5 2A		LDA BNKNUM
CA70: 8D 8FC0		STA OC08F ;SELECT NEXT BANK
CA73: A9 D0		LDA #0D0
CA75: 85 1A		STA PGAD+1
CA77: 4C 59CA		JMP CNTLD
CA7A:		
CA3D*: 3B		
CA7A: 2C 8BC0	PWRITE	BIT OC08B
CA7D: 2C 8BC0		BIT OC08B ;WRITE ENABLE AUX BANK
CA80: A6 27		LDX BYTESTOTRAN+1
CA82: F0**		BEQ PRTWR
CA84: A0 00	PSAVE	LDY #0
CA86: B1 1B	\$1	LDA @TRANADDR,Y ;TRANSFER FULL
CA88: 91 19		STA @PGAD,Y ; 256 BYTE BLOCK
CA8A: C8		INY
CA8B: D0F9		BNE \$1
CA8D: E6 1C		INC TRANADDR+1
CA8F: E6 1A		INC PGAD+1
CA91: A9 E0		LDA #0E0
CA93: C5 1A		CMP PGAD+1
CA95: F0**		BEQ NPCSV
CA97: CA	CNTSV	DEX
CA98: D0EA		BNE PSAVE
CA82*: 16		
CA9A: A6 26	PRTWR	LDX BYTESTOTRAN
CA9C: F0**		BEQ COMPLETE
CA9E: A0 00		LDY #0

```

CAA0: B1 1B      $1    LDA #TRANADDR,Y ;TRANSFER PARTIAL
CAA2: 91 19      STA #PGAD,Y ; 256 BYTE BLOCK
CAA4: C8          INY
CAA5: CA          DEX
CAA6: D0F8        BNE $1
CAA8: F0**        BEQ COMPLETE
CA95* 13
CAA1: C6 2A      NPCSV DEC BNKNUM
CAA1: A5 2A      LDA BNKNUM
CAA1: 8D 8FC0    STA OC08F ;SELECT NEXT BANK
CAB1: A9 D0      LDA #0D0
CAB3: 85 1A      STA PGAD+1
CAB5: 4C 97CA    JMP CNTSV
CAB8:
CAA8* 0E
CA9C* 1A
CA6A* 4C
CA5E* 58
CAB8:           COMPLETE:
CAB8: 20 31C9    JSR RESTOREBNK
CABB: A2 00      LDX #0 ;ASSUME NO ERRORS
CABD: 60          RTS
CABE:
CABE: 0000        SADR :EQU 0
CABE: 0002        DADR :EQU 2
CABE:
CABE:           ;
CABE:           ; boot code on pascal boot mini disc
CABE:           ; can be patched to enter this routine
CABE:           ; with program BPATCH so the app-l-cache
CABE:           ; is initialized before loading the
CABE:           ; pascal interpreter
CABE:
CABE:           ;
CABE:           ;
CABE:           ;
C813* BECA
CABE:           AXBNKINIT:
CABE: A9 3C        LDA #3C
CAC0: 8D 8FC0    STA OC08F
CAC3: 2C 81C0    BIT OC081
CAC6: A0 00        LDY #0
CAC8: B9 ****    LDA CADR,Y
CACB: 85 00        STA SADR
CACD: C8          INY
CACE: B9 ****    LDA CADR,Y
CAD1: 85 01        STA SADR+1 ;get source address of original boot code
CAD3: C8          INY
CAD4: B9 ****    LDA CADR,Y
CAD7: 85 02        STA DADR
CAD9: C8          INY
CADA: B9 ****    LDA CADR,Y
CADD: 85 03        STA DADR+1 ;get address to put this code
CADF: A0 00        LDY #0
CAE1: B1 00        $1    LDA #SADR,Y ;move 1st 12 bytes of the orig. boot code to its
CAE3: 91 02        STA #DADR,Y ;proper location overlaying our intercept patch
CAE5: C8          INY

```

CAE6: C0 0C CPY #0C
CAE8: 90F7 BCC \$1
CAEA: 6C 0200 JMP \$DADR ;jump back to corrected boot code
CAED:
CADB* EDCA
CAD5* EDCA
CACF* EDCA
CAC9* EDCA
CAED: **** CADR ,WORD MBCDE ;address of 1st 12 bytes of mini boot code
CAEF: 0108 ,WORD 801 ;location where mini boot code is loaded
CAF1:
CAED* F1CA
CAF1: E0 60 MBCDE CPX #60
CAF3: F0** BEQ \$1
CAF5: 4C E308 JMP 8E3
CAF3* 03
CAF8: AD 0008 \$1 LDA 800
CAF9: C9 04 CMP #4
CAF1:
CAF1: ,END

PAGE - 12 CONTROL FILE: SYMBOLTABLE DUMP

AB - Absolute LB - Label UD - Undefined MC - Macro
RF - Ref DF - Def PR - Proc FC - Func
PB - Public PV - Private CS - Consts

AXBNKINI	LB CABE:	BFAD	AB 001B:	BLKNUM	AB 0020:	BLKTRN	LB C9FF:	BNKND	AB 001D:	BNKNUM	AB 002A:	BNKSTAT	AB 001I
BYTESTOT	AB 0026:	CADR	LB CAED:	CNTLD	LB CA59:	CNTSV	LB CA97:	COMPLETE	LB CAB8:	CONT	LB C87E:	CONTROL	PR ----
DADR	AB 0002:	DDISK	LB C817:	DELAY	LB C9CE:	DON1	LB C9C9:	DONE	LB C9C8:	D0533	LB C803:	DRAM	LB C824
DRERROR	LB C902:	DRLIMITS	LB C815:	DRVNO	AB 0007:	EHAUX	LB C94F:	ENROM	LB C955:	ENRX	LB C95F:	ERROR	LB C90E
FIN	LB C962:	I0B	AB 0048:	K128	AB 0002:	K192	AB 0001:	K256	AB 0000:	K64	AB 0003:	LIMTAB	LB C86E
LOAD	LB C8E5:	MBCDE	LB CAF1:	NOERROR	LB C8FD:	NPGLD	LB CA6C:	NPGSV	LB CAAA:	NXT1	LB C9BC:	NXTBNK	LB C9BE
PASBOOT	LB C812:	PASCAL	LB C806:	PDISK	LB C9DA:	PGAD	AB 0019:	PLOAD	LB CA46:	PRTRD	LB CASC:	PRTWR	LB C994
PSAVE	LB CA84:	PWRITE	LB CA7A:	RAMSIZE	LB C80F:	REFLG	AB BCFF:	RESTOREB	LB C931:	RSIZE	LB C96D:	RSTBNK	LB C809
RW	LB C8C8:	RWFLC	AB 0022:	SADR	AB 0000:	SAFERSIZ	LB C963:	SAVEBNK	LB C80C:	SAVEBNKS	LB C914:	SCNO	AB 0009
SETUP	LB C977:	STORE	LB C8F6:	SZFLG	AB BCFF:	TEMP1	AB 0019:	TEMP2	AB 001B:	TRANADDR	AB 001B:	TRHO	AB 0008
VERSION	AB BF21:	VLND	AB 0037:	VOLNO	AB 0006:	WRT	LB CABE:	WRTEM	AB 001E:				

PAGE - 13 CONTROL FILE:

Current minimum space is 6953 words

Assembly complete: 531 lines
0 Errors flagged on this Assembly