



```

0000 1: ;-----
0000 2: ;   (General DOS formatter for real.dos v1.0
0000 3: ;
0000 4: ;       Copyright 2005 Intergrated Logic Systems
0000 5: ;       Source Code is Copyright Stephen J. Car
0000 6: ; Please do not share this source!
0000 7: ;-----
0000 8: ;       Real.dos xinit.COM
0000 9: ;       This program is to replace
0000 10: ;       xinit.com
0000 11: ;
0000 12: ;   Code History.   Init design,  August 10, 2005
0000 13: ;       last updated 9/11/2005   sjc
0000 14: ;       working on it again 10/09/2005 sj
0000 15: ;       fixed the boot sector stuff 10/14
0000 16: ;   Added percom table
0000 17: ;   added header section and debug code for multi langu
0000 18: ;
0000 19: ;
0000 20: ;   ~~~~~
0000 21: ;-----
0000 22: ; Notes: o This source code MAY NOT be placed for download
0000 23: ;       o "mea" is a macro that loads the address of the
0000 24: ;       into the pointer specified by the second field.
0000 25: ;-----
0000 26: ; Assembler: MADMAC (tm) ST Cross Assembler (Atari Corp)
0000 27: ;       XASM  st and  IBM VERSIONS
0000 28: ;-----
0000 29: com_file_name:      .macro
0000 30: ;       dc.b  13,"filename.ext",$9b
0000 31: ;       dc.b  12,"Xinit.com  ", $9b
0000 32: ;       .endm
0000 33:
0000 34: ;
0000 35: ;~~~~~
0000 36: ; File revision history. Version Number in hex
0000 37: ;
0000 38: file_ver:          equ  $13
0000 39: ;~~~~~
0000 40: ; the Month in dec   for the first time this revision
0000 41: ;was compiled
0000 42: ;
0000 43: c_month:          equ  6
0000 44: ;~~~~~
0000 45: ; the day in dec     for the first time this revision
0000 46: ;was compiled
0000 47: c_day:            equ  16
0000 48: ;~~~~~
0000 49: ; the year in dec    for the first time this revision
0000 50: ;was compiled
0000 51: c_year:           equ  2010
0000 52: ;~~~~~
0000 53: ; Type of compiler used in program

```

```

0000 54: ;
0000 55: ; 0 = unknown Compiler
0000 56: ; 1 = xasm
0000 57: ; 2 = mac_65
0000 58: ; 3 = basic
0000 59: ; 4 = compiled basic xl
0000 60: ; 5 = C65
0000 61: ; 6 = Action
0000 62: ;
0000 63: ; We can add more as time goes on!
0000 64: ;
0000 65: ;
0000 66: xasm:          equ 1
0000 67: mac_65:         equ 2
0000 68: basic:          equ 3
0000 69: basicxl:        equ 4
0000 70: c65:           equ 5
0000 71: action:         equ 6
0000 72:
0000 73: file_compiler:   equ xasm
0000 74: ;~~~~~
0000 75: ; is this relocatable code ?
0000 76: ; anyother value other than 1 or 2 would be unknown
0000 77: r..yes:          equ 1
0000 78: r..no:           equ 2
0000 79: ;
0000 80: ;~~~~~
0000 81: ; Gotta define if it can be relocated
0000 82: l_relocatable:   equ r..yes
0000 83: ;
0000 84: ;
0000 85: ;~~~~~
0000 86: ; Gotta define if it so I can print what I need
0000 87: r..crl:          equ $7d
0000 88: r..crlf:         equ $9b
0000 89: r..space:        equ $20
0000 90: r..supress:      equ -1
0000 91:
0000 92: l_frstscreenbyte: equ r..supress
0000 93: ;~~~~~
0000 94: ; The language the output file is in.
0000 95: ;
0000 96: ;
0000 97: ; 0 = Undefined
0000 98: ; 1 = English
0000 99: ; 2 = German
0000 100: ;
0000 101: r..language:     equ 1
0000 102: ;~~~~~
0000 103:
0000 104: DRIVE:          equ DUNIT
0000 105:
0000 106:

```

```

0000    107:      .include    equates
0000    108:      .include    globals
0000    109:      .include    macros
0000    110: ;
0000    111: ;
3600    112:      .org    $3600
3600    113: ;
3600    114: win_start:
3600    115: header_info:
3600    116:      .include    header    ; this read
38a7 4c0050 117:      jmp    START_IT    ;did this f
38aa    118:
38aa    119:
4000    120:      .org    $4000
4000    121:
4000    122: Split:
4000    123: #####
4000    124: ;
4000    125: ;      =====
4000    126: ;      Stage 1 boot program
4000    127: ;      =====
4000    128: ;
4000    129:
4000    130: initz1:
4000    131: ;
4000    132: BOOTS:
4000 00    133:      dc.b    $00
4001 03    134:      dc.b    $03
4002 00    135:      dc.b    $00
4003 30    136:      dc.b    $30
4004 e007   137:      cpx    #$07
4006 4c8040 138: w01:      jmp    LOADER
4009       139: ;
4009       140: l4009:
4009 0000   141:      dc.w    0
400b 0000   142:      dc.w    0
400d 0000   143:      dc.w    0
400f 00    144:      dc.b    0
4010 0400   145:      dc.w    4
4012 0000   146:      dc.w    0
4014 0000   147:      dc.w    0
4016 202020 148:      dc.b    "    "
401e 00    149:      dc.b    0
401f 00    150: DENLOA:      dc.b    0
4020 20    151:      dc.b    $20
4021 06    152:      dc.b    6
4022 01    153:      dc.b    1
4023 ff    154:      dc.b    -1
4024 ff    155:      dc.b    -1
4025 00    156:      dc.b    0
4026 00    157:      dc.b    0
4027 00    158:      dc.b    0
4028 0000   159: FIRMAP:      dc.w    0

```

```

402a 00 160:      dc.b $00
402b 00 161:      dc.b $00
402c 00 162:      dc.b $00
402d 0000 163:      dc.w 0
402f 00 164:      dc.b $00
4030 165:
4030 166:
4030 457272 167: NODMSG:      dc.b "Error: No DOS",$9b
403e 168: ;
403e 169: ;
403e 170: ; get next sector map.. initz new map #
403e 171: ; -----
403e 172: ;
403e 173:
403e 174: LOANM:
403e ad2840 175: w02:      lda FIRMAP
4041 8d0a03 176:      sta SECTOR
4044 ad2940 177: w03:      lda FIRMAP+1
4047 8d0b03 178:      sta SECTOR+1
404a a900 179:      lda # low secmap
404c a22f 180:      ldx # high secmap
404e 20f140 181: w04:      jsr READS
4051 ad002f 182:      lda secmap
4054 8d2840 183: w05:      sta FIRMAP
4057 ad012f 184:      lda secmap+1
405a 8d2940 185: w06:      sta FIRMAP+1
405d a004 186:      ldy #4
405f 8491 187:      sty SECPTR
4061 188: ;
4061 189: ; get next sector number..
4061 190: ; -----
4061 191: ;
4061 192:
4061 a491 193: GNXSEC:      ldy SECPTR
4063 cc1f40 194: w07:      cpy DENLOA
4066 f0d6 195:      beq LOANM
4068 b9002f 196:      lda secmap,Y
406b 8d0a03 197:      sta SECTOR
406e b9012f 198:      lda secmap+1,Y
4071 8d0b03 199:      sta SECTOR+1
4074 c8 200:      INY
4075 c8 201:      INY
4076 8491 202:      sty SECPTR
4078 60 203: LRTS:      RTS
4079 204: ;
4079 6ce202 205: DOINITZ:  jmp ($2e2)
407c 206: ;
407c 207: 1407c:
407c 00 208:      dc.b 0
407d 00 209:      dc.b 0
407e 00 210:      dc.b 0
407f 00 211:      dc.b 0
4080 212:

```

```

4080 a200 213: LOADER:      ldx  #0
4082 ad1f40 214: w08:      lda  DENLOA          ; s
4085 8591 215:          sta  SECPTR          ;
4087 8590 216:          sta  CHPTR          ;
4089 8d0803 217:          sta  DBYTLO          ; s
408c d001 218:          bne  ISX
408e e8 219:          INX
408f 8e0903 220: ISX:      stx  DBYTHI          ; s
4092 200a41 221: w09:      jsr  GETBYTE          ; g
4095 8596 222:          sta  temp
4097 200a41 223: w10:      jsr  GETBYTE
409a 2596 224:          and  temp
409c c9ff 225:          cmp  #$FF
409e d037 226:          bne  TYPER          ; N
40a0      227: BEGIN:
40a0 a978 228: lb_1:      lda  # low LRTS          ; n
40a2 8de202 229:          sta  INITAD
40a5 a940 230: hb_1:      lda  # high LRTS
40a7 8de302 231:          sta  INITAD+1
40aa 200a41 232: w11:      jsr  GETBYTE
40ad 8592 233:          sta  buf          ; g
40af 200a41 234: w12:      jsr  GETBYTE
40b2 8593 235:          sta  buf+1
40b4 0592 236:          ora  buf
40b6 f01c 237:          beq  STAD          ; a
40b8 200a41 238: w13:      jsr  GETBYTE
40bb 38 239:          SEC
40bc e592 240:          sbc  buf          ; c
40be 48 241:          PHA
40bf 08 242:          PHP
40c0 200a41 243: w14:      jsr  GETBYTE
40c3 28 244:          PLP
40c4 e593 245:          sbc  buf+1
40c6 8595 246:          sta  LENG+1          ; s
40c8 68 247:          PLA
40c9 8594 248:          sta  LENG
40cb      249:
40cb 206b41 250: w15:      jsr  LOABUF          ;
40ce 207940 251: w16:      jsr  DOINITZ          ; r
40d1 4ca040 252: w17:      jmp  BEGIN          ; d
40d4      253: ;
40d4 6ce002 254: STAD:      jmp  ($2e0)
40d7      255: ;
40d7      256: ;
40d7      257: ;
40d7      258: ; error.. no DOS on diskette
40d7      259: ; -----
40d7      260: ;
40d7      261: TYPER:
40d7 a930 262: lb_2:      lda  # low NODMSG
40d9 a240 263: hb_2:      ldx  # high NODMSG
40db 8d4403 264:          sta  ICBAL
40de 8e4503 265:          stx  ICBAH

```

```

40e1 8e4803 266:      stx  ICBL
40e4 a909 267:      lda  #9
40e6 8d4203 268:      sta  ICCOM
40e9 a200 269:      ldx  #0
40eb 2056e4 270:      jsr  CIO
40ee      271:
40ee      272: FOREV:
40ee 4cee40 273: w18:      jmp  FOREV          ; e
40f1      274: ;
40f1      275: ;
40f1      276: ;
40f1      277: ;      read sector at address
40f1      278: ;      -----
40f1      279: ;
40f1      280: READS:
40f1 a040 281:      ldy  #$40          ; r
40f3 8c0303 282:      sty  DSTATS
40f6 8d0403 283:      sta  DBUFLO
40f9 8e0503 284:      stx  DBUFHI          ; n
40fc ad0a03 285:      lda  SECTOR
40ff 0d0b03 286:      ora  SECTOR+1
4102 f0d3 287:      beq  TYPER          ; i
4104 2059e4 288:      jsr  SIO            ; u
4107 30ce 289:      bmi  TYPER          ; i
4109 60 290:      RTS
410a      291: ;
410a      292: ;
410a      293: ;      get byte/burst routine
410a      294: ;      -----
410a      295: ;
410a      296: ;
410a      297: GETBYTE:
410a a900 298:      lda  #0
410c 8595 299:      sta  LENG+1
410e 8594 300:      sta  LENG
4110      301:
4110 a690 302: GETBYT:      ldx  CHPTR
4112 ec1f40 303: w19:      cpx  DENLOA          ; c
4115 f006 304:      beq  LOANS          ;
4117 bd002e 305:      lda  datbuf,X
411a e690 306:      inc  CHPTR
411c 60 307:      RTS
411d      308: ;
411d      309: ;
411d      310: ;
411d      311: ;      Load next sector in buffer/ or burst if possible
411d      312: ;      -----
411d      313: ;
411d      314: LOANS:
411d 206140 315: w20:      jsr  GNXSEC          ; g
4120 a595 316:      lda  LENG+1          ; c
4122 d017 317:      bne  OKBUR
4124 ad1f40 318: w21:      lda  DENLOA

```

```

4127 f004 319:      beq  NOBUR
4129 a594 320:      lda  LENG
412b 300e 321:      bmi  OKBUR          ; i
412d      322:
412d a900 323: NOBUR:  lda  #low datbuf
412f a22e 324:      ldx  #high datbuf
4131 20f140 325: w22:  jsr  READS          ; r
4134 38 326:      SEC
4135 2690 327:      rol  CHPTR          ; p
4137 ad002e 328:      lda  datbuf          ; g
413a 60 329:      RTS
413b      330: ;
413b      331: ;
413b      332: ; Do burst read sector
413b      333: ; -----
413b      334: ;
413b a592 335: OKBUR:  lda  buf          ; g
413d a693 336:      ldx  buf+1
413f 20f140 337: w23:  jsr  READS          ; r
4142 a592 338:      lda  buf          ; a
4144 18 339:      CLC
4145 6d0803 340:      adc  DBYTLO
4148 8592 341:      sta  buf
414a a593 342:      lda  buf+1
414c 6d0903 343:      adc  DBYTHI
414f 8593 344:      sta  buf+1
4151 38 345:      SEC          ; s
4152 a594 346:      lda  LENG
4154 ed0803 347:      sbc  DBYTLO
4157 8594 348:      sta  LENG
4159 a595 349:      lda  LENG+1
415b ed0903 350:      sbc  DBYTHI
415e 8595 351:      sta  LENG+1
4160 4c1d41 352: w24:  jmp  LOANS          ; g
4163      353: ;
4163      354: ;
4163      355: ;
4163      356: ; Load entire buffer from file
4163      357: ; -----
4163      358: ;
4163      359: ;
4163 a594 360: LOABUFX:  lda  LENG          ; d
4165 d002 361:      bne  NOB
4167 c695 362:      dec  LENG+1
4169 c694 363: NOB:    dec  LENG
416b      364:
416b      365:
416b      366: LOABUF:
416b 201041 367: w25:  jsr  GETBYT          ; g
416e a000 368:      ldy  #0
4170 9192 369:      sta  (buf),Y
4172 e692 370:      inc  buf
4174 d002 371:      bne  SK11

```



```

4176 e693 372:      inc  buf+1
4178 a594 373: SK11:   lda  LENG
417a 0595 374:      ora  LENG+1
417c d0e5 375:      bne  LOABUFX
417e 60   376:      RTS
417f 00   377:      brk
4180      378:
4180      379: zend1:  ds.b  0
4180      380:
4180      381:
4180      382: BOOTI:  =  SPLIT                      ; l
4180      383:
4180      384: ;
4180      385: ;
4180      386: ;
5000      387:      .org $5000
5000      388: ;=====
5000      389: ;   Beginning and initialization
5000      390: ;=====
5000      391: ;
5000      392: START_IT:
5000 200b60 393:      jsr  rlocate
5003 a50a 394: BEGINI:   lda  COMTAB
5005 38   395:      SEC
5006 e90a 396:      sbc  # low lsio
5008 8d1650 397:      sta  XSIO+1
500b a50b 398:      lda  COMTAB+1
500d e900 399:      sbc  # high lsio
500f 8d1750 400:      sta  XSIO+2
5012 4c1850 401:      jmp  FORMAT
5015 6c0000 402: XSIO:    jmp  (0)
5018      403: ;
5018      404: ;=====
5018      405: ;   Main parameter input
5018      406: ;=====
5018 ba   407: FORMAT:   TSX
5019 8e465a 408:      STX  STACKP
501c 20a438 409:      jsr  end_of_header-3
501f      410:
501f      411:
501f      412: START:
501f      413: ;
501f      414: ;
501f      415: ; GET DRIVE FOR DIRECTORY
501f      416: ;
501f a50a 417:      LDA  COMTAB                      ; g
5021 18   418:      CLC
5022 6903 419:      ADC  # low ZCRNAME
5024 8d2f50 420:      STA  JVC+1
5027 a50b 421:      LDA  COMTAB+1
5029 6900 422:      ADC  # high ZCRNAME
502b 8d3050 423:      STA  JVC+2
502e 200000 424: JVC:     JSR  $00                      ; a

```

```

5031      425: ;
5031      426: ;
5031 a021 427:      LDY  #$21                ; D
5033 b10a 428: MVNA:      LDA  (COMTAB),Y
5035 998759 429:      STA  FSPEC-$21,Y
5038 c8   430:      INY
5039 c024 431:      CPY  #$24
503b d0f6 432:      BNE  MVNA
503d      433: ;
503d      434: ;
503d      435: ; DO DIRECTORY FOR .DOS FILES
503d      436: ; -----
503d a900 437: DODIR:      LDA  #0                ; z
503f 8d485a 438:      STA  DOSP
5042 8d535a 439:      STA  DIRP
5045      440:
5045 a210 441:      IDX  #$10
5047 a903 442:      lda  #3
5049 9d4203 443:      sta  ICCOM,X
504c a9a8 444:      lda  # low FSPEC
504e 9d4403 445:      sta  ICBAL,X
5051 a959 446:      lda  # high FSPEC
5053 9d4503 447:      sta  ICBAH,X
5056 a906 448:      lda  #6
5058 9d4a03 449:      sta  ICAX1,X
505b a900 450:      lda  #0
505d 9d4b03 451:      sta  ICAX2,X
5060 2056e4 452:      jsr  CIO
5063 1037 453:      bpl  OKO
5065      454: ;
5065 201957 455: ERROR:      jsr  printsi
5068 9b4572 456:      dc.b  $9B,"Error #",-1
5071 98   457:      TYA
5072 204b57 458:      jsr  ECHO_HEX
5075 201957 459:      jsr  printsi
5078 202d20 460:      dc.b  " - Format Aborted..",$9B,-1
508d ae465a 461: RETURN:      ldx  STACKP
5090 9a   462:      TXS
5091 a90c 463: CLOSE:      lda  #CCCLoS                ; c
5093 a210 464:      ldx  #$10
5095 9d4203 465:      sta  ICCOM,X
5098 2056e4 466:      jsr  CIO
509b 60   467:      RTS
509c      468: ;
509c a905 469: OKO:      lda  #5                ; g
509e a210 470:      ldx  #$10
50a0 9d4203 471:      sta  ICCOM,X
50a3 a9d6 472:      lda  # low ENTRY
50a5 9d4403 473:      sta  ICBAL,X
50a8 a95a 474:      lda  # high ENTRY
50aa 9d4503 475:      sta  ICBAH,X
50ad a928 476:      lda  #40
50af 9d4803 477:      sta  ICBLL,X

```

```

50b2 2056e4 478:      jsr  CIO
50b5 30ae 479:      bmi  ERROR
50b7 480: ;
50b7 add65a 481:      lda  ENTRY          ; a
50ba c930 482:      cmp  #'0'          ;
50bc b036 483:      bcs  XDIR
50be 484: ;
50be ac535a 485:      ldy  DIRP          ; g
50c1 ae485a 486:      ldx  DOSP          ; p
50c4 98 487:      TYA
50c5 9d495a 488:      sta  DOSPT,X
50c8 e8 489:      INX
50c9 8e485a 490:      stx  DOSP
50cc 491: ;
50cc a202 492:      ldx  #2          ; n
50ce bdd65a 493: MVNA1:  lda  ENTRY,X
50d1 c920 494:      cmp  #$20
50d3 f00b 495:      beq  ATED
50d5 e00a 496:      cpx  #10
50d7 f007 497:      beq  ATED
50d9 99545a 498:      sta  NAMES,Y
50dc e8 499:      INX
50dd c8 500:      INY
50de d0ee 501:      bne  MVNA1
50e0 502: ;
50e0 a200 503: ATED:  ldx  #0          ; m
50e2 bdac59 504: MVNA2:  lda  EXTN,X
50e5 99545a 505:      sta  NAMES,Y
50e8 c8 506:      INY
50e9 e8 507:      INX
50ea e005 508:      cpx  #5
50ec d0f4 509:      bne  MVNA2
50ee 8c535a 510:      sty  DIRP          ; s
50f1 4c9c50 511:      jmp  OKO
50f4 512: ;
50f4 209150 513: XDIR:  jsr  CLOSE          ; c
50f7 514: ;
50f7 ad485a 515:      lda  DOSP          ; #
50fa d031 516:      bne  SOME
50fc a2ff 517:      ldx  #-1
50fe 8e475a 518:      stx  CURP
5101 201957 519:      jsr  printsi
5104 4e6f20 520:      dc.b  "No "
5107 22444f 521:      dc.b  $22,"DOS",$22
510c 204669 522:      dc.b  " Files Found on Diskette...",$9B,$
512a 523: ;
512a 4cc651 524:      jmp  CONTIN
512d 525: ;
512d 526: ; GET DOS IMAGE SELECTION
512d 527: ; -----
512d a200 528: SOME:  ldx  #0
512f 201957 529:      jsr  printsi
5132 53656c 530:      dc.b  "Select Your DOS Version:",$9B,$9B,

```

```

514d 8e455a 531: SAP:      stx  COUNT          ; g
5150 bc495a 532:      ldy  DOSPT,X
5153 201957 533:      jsr  printsi
5156 202020 534:      dc.b  " ",-1
515a e8     535:      INX
515b 8a     536:      TXA          ; p
515c 206b57 537:      jsr  PRNIB          ;
515f 201957 538:      jsr  printsi
5162 292020 539:      dc.b  ") ",-1
5166 b9545a 540: LPCH:      lda  NAMES,Y          ; p
5169 c92e   541:      cmp  #'.'          ;
516b f006   542:      beq  ENAN
516d 200057 543:      jsr  ECHO
5170 c8     544:      INY
5171 d0f3   545:      bne  LPCH
5173 201957 546: ENAN:      jsr  printsi
5176 9bff   547:      dc.b  $9B,-1
5178 ae455a 548:      ldx  COUNT
517b e8     549:      INX          ; g
517c ec485a 550:      cpx  DOSP
517f d0cc   551:      bne  SAP
5181       552: ;
5181 201957 553:      jsr  printsi
5184 202020 554:      dc.b  " N) -No DOS-",$9B
5194 9b4368 555:      dc.b  $9B,"Choice ? ",-1
519f 20f556 556: LPN:      jsr  INCH
51a2 c94e   557:      cmp  #'N'
51a4 d007   558:      bne  CKNUM
51a6 a2ff   559:      ldx  #-1
51a8 8e475a 560:      stx  CURP
51ab d010   561:      bne  SHOWN
51ad c931   562: CKNUM:      cmp  #'1'
51af 90ee   563:      bcc  LPN
51b1 e931   564:      sbc  #'1'
51b3 cd485a 565:      cmp  DOSP
51b6 b0e7   566:      bcs  LPN
51b8 8d475a 567:      sta  CURP
51bb 6931   568:      adc  #'1'
51bd 200057 569: SHOWN:      jsr  ECHO
51c0 201957 570:      jsr  printsi
51c3 9b9bff 571:      dc.b  $9B,$9B,-1
51c6       572: ;
51c6       573: ; READ DOS IMAGE
51c6       574: ;
51c6 2c475a 575: CONTIN:      bit  CURP
51c9 306c   576:      bmi  NOREA          ; i
51cb       577: ;
51cb ae475a 578:      ldx  CURP          ; g
51ce bc495a 579:      ldy  DOSPT,X
51d1 a200   580:      ldx  #0
51d3 b9545a 581: MVNAM3:      lda  NAMES,Y
51d6 9df25a 582:      sta  DOSI+3,X
51d9 c99b   583:      cmp  #$9B

```

```

51db f004 584:      beq  ATE2
51dd e8 585:      INX
51de c8 586:      INY
51df d0f2 587:      bne  MVNAM3
51e1 a202 588: ATE2:   ldx  #3-1
51e3 bda859 589: MVNAM4:   lda  FSPEC,X
51e6 9def5a 590:      sta  DOSI,X
51e9 ca 591:      DEX
51ea 10f7 592:      bpl  MVNAM4
51ec 593: ;
51ec a903 594:      lda  #3              ; o
51ee a210 595:      ldx  #$10
51f0 9d4203 596:      sta  ICCOM,X
51f3 a9ef 597:      lda  # low DOSI
51f5 9d4403 598:      sta  ICBAL,X
51f8 a95a 599:      lda  # high DOSI
51fa 9d4503 600:      sta  ICBAH,X
51fd a904 601:      lda  #4
51ff 9d4a03 602:      sta  ICAX1,X          ; o
5202 2056e4 603:      jsr  CIO              ; a
5205 1003 604:      bpl  OKRD2
5207 4c6550 605: ERJX1:   jmp  ERROR
520a 606: ;
520a a907 607: OKRD2:   lda  #CCGTCH          ; r
520c 9d4203 608:      sta  ICCOM,X
520f a909 609:      lda  # low TABLE      ; p
5211 9d4403 610:      sta  ICBAL,X
5214 a95f 611:      lda  # high TABLE
5216 9d4503 612:      sta  ICBAH,X
5219 a980 613:      lda  #128              ; 3
521b 9d4803 614:      sta  ICBLL,X
521e 9d4903 615:      sta  ICBLH,X
5221 2056e4 616:      jsr  CIO
5224 c088 617:      cpy  #$88
5226 d0df 618:      bne  ERJX1
5228 bd4803 619:      lda  ICBLL,X          ; G
522b 8d075b 620:      sta  TABLEN
522e bd4903 621:      lda  ICBLH,X
5231 8d085b 622:      sta  TABLEN+1
5234 623: ;
5234 209150 624:      jsr  CLOSE          ; c
5237 625: ;
5237 a03d 626: NOREA:   ldy  #60+1
5239 a942 627:      lda  # low INIT
523b 910a 628:      sta  (COMTAB),Y
523d c8 629:      INY
523e a952 630:      lda  # high INIT
5240 910a 631:      sta  (COMTAB),Y
5242 632: ;
5242 633: ;
5242 634: ;
5242 635: ; GET DRIVE TO FORMAT
5242 636: ; -----

```

```

5242 ba 637: INIT:      TSX
5243 8e465a 638:      stx  STACKP
5246      639: ;
5246 ea 640: .a:      nop
5247 ea 641:      nop
5248 ea 642:      nop
5249 a920 643:      lda  #$20
524b 8d4652 644:      sta  .a
524e a9a4 645:      lda  #low_end_of_header-3
5250 8d4752 646:      sta  .a+1
5253 a938 647:      lda  #high_end_of_header-3
5255 8d4852 648:      sta  .a+2
5258      649:
5258 201957 650:      jsr  printsi          ; d
525b 9b9b44 651:      dc.b  $9B,$9B,"Drive to format ?",-1
5270 20be56 652:      jsr  INDRIV
5273 e92f 653:      sbc  #47              ; c
5275 8d0103 654:      sta  DUNIT              ; s
5278      655: ;
5278      656: ;
5278      657: ;
5278      658: ; GET TRACKS/SIDES
5278      659: ;
5278 201957 660:      jsr  printsi
527b 9b5365 661:      dc.b  $9B,"Select number of tracks:",$9B,
5296 203129 662:      dc.b  " 1) 40 trks/SS  5) 40 trks/DS",$9
52b5 203229 663:      dc.b  " 2) 77 trks/SS  6) 77 trks/DS",$9
52d4 203329 664:      dc.b  " 3) 80 trks/SS  7) 80 trks/DS",$9
52f3 203429 665:      dc.b  " 4) 35 trks/SS  8) 35 trks/DS",$9
5312      666: ;      dc.b  " 9) Format ramdisk 512",$9b
5312 9b4368 667:      dc.b  $9B,"Choice ?",-1
531d 20f556 668: REIN:      jsr  INCH
5320 c931 669:      cmp  #'1'
5322 90f9 670:      bcc  REIN
5324 c93a 671:      cmp  #'9'+1          ; g
5326 b0f5 672:      bcs  REIN
5328 200057 673:      jsr  ECHO
532b 290f 674:      and  #15
532d aa 675:      TAX
532e ca 676:      DEX
532f bdf059 677:      lda  TRKST,X          ; g
5332 8d425a 678:      sta  TRACKS
5335 a000 679:      ldy  #0
5337 297f 680:      and  #$7F
5339 c94d 681:      cmp  #77
533b d001 682:      bne  PVAL
533d 88 683:      DEY
533e 8c415a 684: PVAL:      sty  INCH8
5341      685: ;
5341 8a 686:      TXA              ; g
5342 0a 687:      ASL
5343 8d435a 688:      sta  TRKINX
5346 0a 689:      ASL

```

```

5347 6d435a 690:      adc  TRKINX
534a 8d435a 691:      sta  TRKINX
534d      692: ;
534d      693: ;
534d      694: ; GET DENSITY/LENGTH/#SECTORS
534d      695: ;
534d 201957 696:      jsr  printsi
5350 9b9b53 697:      dc.b  $9B,$9B,"Select density:",$9B,$9B
5363 203129 698:      dc.b  " 1) Single density",$9B
5376 203229 699:      dc.b  " 2) Double density",$9B,-1
538a 2c415a 700:      bit  INCH8
538d 301c 701:      bmi  NO1050
538f 201957 702:      jsr  printsi
5392 203329 703:      dc.b  " 3) 1050 double density",$9B,-1
53ab 201957 704: NO1050:      jsr  printsi
53ae 9b4368 705:      dc.b  $9B,"Choice ?",-1
53b9 20f556 706: REIN2:      jsr  INCH
53bc c931 707:      cmp  #'1'
53be 90f9 708:      bcc  REIN2
53c0 c934 709:      cmp  #'4'
53c2 b0f5 710:      bcs  REIN2
53c4 2c415a 711:      bit  INCH8
53c7 1004 712:      bpl  OKCH2
53c9 c933 713:      cmp  #'3'
53cb f0ec 714:      beq  REIN2
53cd 200057 715: OKCH2:      jsr  ECHO
53d0 2907 716:      and  #7
53d2 8d445a 717:      sta  DENINX
53d5 aa 718:      TAX
53d6 2c415a 719:      bit  INCH8
53d9 3014 720:      bmi  GET8V
53db      721: ;
53db bde659 722:      lda  LEN5T-1,X          ; g
53de 8d3c5a 723:      sta  DENSITY
53e1 bde959 724:      lda  DEN5T-1,X          ; g
53e4 8d405a 725:      sta  DENS
53e7 bdec59 726:      lda  SEC5T-1,X
53ea 8d3d5a 727:      sta  SECTORS
53ed d012 728:      bne  CON
53ef      729: ;
53ef bde059 730: GET8V:      lda  LEN8T-1,X          ; g
53f2 8d3c5a 731:      sta  DENSITY
53f5 bde259 732:      lda  DEN8T-1,X          ; g
53f8 8d405a 733:      sta  DENS
53fb bde459 734:      lda  SEC8T-1,X
53fe 8d3d5a 735:      sta  SECTORS
5401      736: ;
5401 ca 737: CON:      DEX
5402 8a 738:      TXA
5403 0a 739:      ASL
5404 6d435a 740:      adc  TRKINX
5407 aa 741:      TAX          ; g
5408 bdb159 742:      lda  MAXST,X

```

```

540b 8d3e5a 743:      sta  MAXSEC
540e bdb259 744:      lda  MAXST+1,X
5411 8d3f5a 745:      sta  MAXSEC+1
5414      746: ;
5414      747: ;
5414      748: ; GET VOLUME NAME
5414      749: ;
5414 201957 750: GETVOL:      jsr  printsi
5417 9b9b56 751:      dc.b  $9B,$9B,"Volume Name ? ",-1
5428 a008 752:      ldy  #8
542a a916 753:      lda  #BMD+SDVOL
542c a240 754:      ldx  # high BOOTI
542e 20c657 755:      jsr  GETLINE
5431      756: ;
5431      757: ;
5431      758: ; SEE IF TO FORMAT IN 'US'
5431      759: ;
5431 201957 760: OKVOL:      jsr  printsi
5434 9b9b55 761:      dc.b  $9B,$9B,"UltraSpeed Sector Skew (Do
5459 20d256 762:      jsr  INYN
545c 6eff5a 763:      ror  US              ; u
545f      764: ;
545f      765: ;
545f      766: ; SEE IF OTHER HIGH SPEED SKEW
545f 2cff5a 767:      bit  US
5462 102d 768:      bpl  ISUS1
5464 201957 769:      jsr  printsi
5467 9b496e 770:      dc.b  $9B,"Indus or XF551 High Speed Skew
548b 20d256 771:      jsr  INYN
548e 6e005b 772:      ror  HIGHSP
5491      773: ISUS1:
5491      774: ; FORMAT DISKETTE
5491      775: ;
5491 201957 776:      jsr  printsi
5494 9b496e 777:      dc.b  $9B,"Insert Diskette to Format,",$9
54b0 507265 778:      dc.b  "Press <RETURN> When Ready.. ",$9B,
54ce 20f556 779:      jsr  INCH
54d1      780: ;
54d1      781: ;
54d1 201957 782:      jsr  printsi
54d4 9b9b46 783:      dc.b  $9B,$9B,"Formatting...",-1
54e4      784: ;
54e4 a909 785:      lda  # low FMTBUF
54e6 8d0403 786:      sta  DBUFLO
54e9 a95b 787:      lda  # high FMTBUF
54eb 8d0503 788:      sta  DBUFHI
54ee a931 789:      lda  #$31
54f0 8d0003 790:      sta  DDEVIC
54f3      791: ;
54f3 a953 792:      lda  #'S'
54f5 8d0203 793:      sta  DCOMND
54f8 a940 794:      lda  #$40
54fa 8d0303 795:      sta  DSTATS              ; g

```



```

54fd a904 796:      lda  #4          ;
54ff 8d0803 797:      sta  DBYTLO
5502 a900 798:      lda  #0
5504 8d0903 799:      sta  DBYTHI
5507 201550 800:      jsr  XSIO
550a      801: ;
550a a94e 802:      lda  #'N'          ; g
550c 8d0203 803:      sta  DCOMND
550f a940 804:      lda  #$40
5511 8d0303 805:      sta  DSTATS
5514 a90c 806:      lda  #12
5516 8d0803 807:      sta  DBYTLO
5519 a900 808:      lda  #0
551b 8d0903 809:      sta  DBYTHI
551e 201550 810:      jsr  XSIO
5521 3077 811:      bmi  NOTCONF      ; j
5523      812: ;
5523 a94f 813:      lda  #'O'
5525 a20c 814:      ldx  #12
5527 2cff5a 815:      bit  US
552a 3004 816:      bmi  SETTY          ; j
552c a966 817:      lda  #'f'
552e a280 818:      ldx  #128          ; n
5530      819: ;
5530 8d0203 820: SETTY:  sta  DCOMND      ; s
5533 8e0803 821:      stx  DBYTLO          ;
5536 a900 822:      lda  #0
5538 8d0903 823:      sta  DBYTHI
553b a980 824:      lda  #$80
553d 8d0303 825:      sta  DSTATS
5540      826: ;
5540 ad425a 827:      lda  TRACKS          ; s
5543 297f 828:      and  #$7F
5545 8d095b 829:      sta  FMTBUF+0
5548 ad3d5a 830:      lda  SECTORS          ; s
554b 8d0c5b 831:      sta  FMTBUF+3
554e a200 832:      ldx  #0
5550 ad3c5a 833:      lda  DENSITY
5553 8d105b 834:      sta  FMTBUF+7          ; s
5556 3002 835:      bmi  ISSN
5558 a201 836:      ldx  #1
555a 8e0f5b 837: ISSN:  stx  FMTBUF+6
555d ad405a 838:      lda  DENS          ; s
5560 8d0e5b 839:      sta  FMTBUF+5
5563 ad425a 840:      lda  TRACKS
5566 2980 841:      and  #$80
5568 0a 842:      ASL
5569 2a 843:      ROL
556a 8d0d5b 844:      sta  FMTBUF+4          ; s
556d      845: ;
556d ae445a 846:      ldx  DENINX          ; g
5570 ca 847:      DEX
5571 8a 848:      TXA

```

```

5572 0a 849:      ASL                      ; s
5573 aa 850:      TAX                      ;
5574 bdf859 851:    lda  ORDT,X
5577 8d8355 852:    sta  GVAL+1
557a bdf959 853:    lda  ORDT+1,X
557d 8d8455 854:    sta  GVAL+2
5580 a200 855:    ldx  #0
5582 bdffff 856: GVAL:    lda  $FFFF,X
5585 9d155b 857:    sta  FMTBUF+12,X
5588 e8 858:      INX
5589 e01a 859:    cpx  #26
558b d0f5 860:    bne  GVAL
558d      861: ;
558d 2cff5a 862:    bit  US                      ; c
5590 106b 863:    bpl  FMTWRM                  ; j
5592 201550 864:    jsr  XSIO                      ; s
5595 103a 865:    bpl  STDFMT                  ;
5597 4c6550 866:    jmp  ERROR
559a      867: ;
559a a9ff 868: NOTCONF:    lda  #$FF                      ; c
559c 8d005b 869:    sta  HIGHSP
559f 2cff5a 870:    bit  US                      ; c
55a2 3023 871:    bmi  NOUS
55a4 201957 872:    jsr  printsi
55a7 9b4472 873:    dc.b  $9B,"Drive Cannot Run UltraSpeed.."
55c7      874: ;
55c7 ad445a 875: NOUS:    lda  DENINX
55ca c903 876:    cmp  #3
55cc d003 877:    bne  STDFMT
55ce a222 878:    ldx  #$22                      ; 1
55d0 2c 879:      dc.b  $2C
55d1 a221 880: STDFMT:    ldx  #$21
55d3 2c005b 881:    bit  HIGHSP
55d6 3002 882:    bmi  STD2                      ; j
55d8 a2a1 883:    ldx  #$A1
55da 8e0203 884: STD2:    stx  DCOMND
55dd      885: ;
55dd ae3c5a 886:    ldx  DENSITY
55e0 8e0803 887:    stx  DBYTLO
55e3 a900 888:    lda  #0
55e5 e000 889:    cpx  #0
55e7 d002 890:    bne  LLK
55e9 a901 891:    lda  #1
55eb 8d0903 892: LLK:    sta  DBYTHI
55ee a940 893:    lda  #$40
55f0 8d0303 894:    sta  DSTATS
55f3 a904 895:    lda  #4
55f5 8d0a03 896:    sta  SECTOR
55f8 a900 897:    lda  #0
55fa 8d0b03 898:    sta  SECTOR+1
55fd      899: ;
55fd a9f8 900: FMTWRM:    lda  #$F8
55ff 8d0603 901:    sta  DTIMLO

```

```

5602      902: ;
5602 201550 903: NOSPEC:      jsr  XSIO
5605 1003  904:      bpl  OKFMT2      ;j
5607 4c6550 905: XERR:      jmp  ERROR
560a      906: ;
560a 202259 907: OKFMT2:      jsr  RECALC
560d 201158 908:      jsr  WRITE      ; w
5610      909: ;
5610 2c475a 910:      bit  CURP      ; c
5613 306a  911:      bmi  NOMS1
5615 201957 912:      jsr  printsi
5618 9b9b57 913:      dc.b  $9B,$9B,"Writing your DOS Selection
5636      914: ;
5636 a903  915:      lda  #CCOPEN
5638 a210  916:      ldx  #$10
563a 9d4203 917:      sta  ICCOM,X
563d a9ef  918:      lda  # low DOSI
563f 9d4403 919:      sta  ICBAL,X
5642 a95a  920:      lda  # high DOSI
5644 9d4503 921:      sta  ICBAH,X
5647 ad0103 922:      lda  dunit
564a      923: ;
564a 6930  924:      adc  #48
564c 8df05a 925:      sta  DOSI+1
564f a908  926:      lda  #8
5651 9d4a03 927:      sta  ICAX1,X
5654 2056e4 928:      jsr  CIO      ; o
5657 1003  929:      bpl  WRTDN
5659 4c6550 930: JERX2:      jmp  ERROR
565c      931: ;
565c a90b  932: WRTDN:      lda  #CCPUCH      ; p
565e 9d4203 933:      sta  ICCOM,X
5661 a909  934:      lda  # low TABLE
5663 9d4403 935:      sta  ICBAL,X
5666 a95f  936:      lda  # high TABLE
5668 9d4503 937:      sta  ICBAH,X
566b ad075b 938:      lda  TABLEN
566e 9d4803 939:      sta  ICBLH,X
5671 ad085b 940:      lda  TABLEN+1
5674 9d4903 941:      sta  ICBLH,X
5677 2056e4 942:      jsr  CIO      ; w
567a 30dd  943:      bmi  JERX2
567c 209150 944:      jsr  CLOSE      ; c
567f      945: ;
567f 201957 946: NOMS1:      jsr  printsi
5682 9b9b44 947:      dc.b  $9B,$9B,"Diskette Initialized...",$
569c 507265 948:      dc.b  "Press <RETURN> to Continue",-1
56b8 20f556 949:      jsr  INCH
56bb 4c4252 950:      jmp  INIT      ; c
56be      951: ;
56be      952: ;
56be      953: ;
56be      954: ;=====

```

```

56be 955: ; Input subroutines
56be 956: ;=====
56be 957: ;
56be 958: ; Get drive number
56be 959: ; -----
56be 960: ;
56be 20f556 961: INDRIV: jsr INCH
56c1 c931 962: cmp #'1'
56c3 90f9 963: bcc INDRIV
56c5 c93a 964: cmp #'9'+1 ;'9
56c7 b0f5 965: bcs INDRIV
56c9 200057 966: jsr ECHO
56cc 201957 967: jsr printsi
56cf 9bff 968: dc.b $9B,-1
56d1 60 969: RTS
56d2 970: ;
56d2 971: ;
56d2 972: ; Get YES/NO response.. CS if NO
56d2 973: ; -----
56d2 974: ;
56d2 20f556 975: INYN: jsr INCH
56d5 c959 976: cmp #'Y'
56d7 f014 977: beq YES
56d9 c979 978: cmp #'y'
56db f010 979: beq YES
56dd c94e 980: cmp #'N'
56df f004 981: beq NO
56e1 c96e 982: cmp #'n'
56e3 d0ed 983: bne INYN
56e5 201957 984: NO: jsr printsi
56e8 4e9bff 985: dc.b "N",$9B,-1
56eb 38 986: SEC
56ec 60 987: RTS
56ed 201957 988: YES: jsr printsi
56f0 599bff 989: dc.b "Y",$9B,-1
56f3 18 990: CLC
56f4 60 991: RTS
56f5 992: ;
56f5 993: ;
56f5 994: ; Input character with escape processing
56f5 995: ; -----
56f5 996: ;
56f5 20ab57 997: INCH: jsr GET_KEY
56f8 c91b 998: cmp #27
56fa f001 999: beq EXI
56fc 60 1000: RTS
56fd 4c8d50 1001: EXI: jmp RETURN
5700 1002: ;
5700 1003: ;
5700 1004: ;
5700 1005: ;=====
5700 1006: ; print/input routines
5700 1007: ;=====

```

```

5700      1008: ;
5700      1009: ;   Print character
5700      1010: ;   -----
5700      1011: ; in:
5700      1012: ;   A   = character to print
5700      1013: ; out:
5700      1014: ;   all registers preserved
5700      1015: ;
5700 207457 1016: ECHO:      jsr  SAVER
5703 200957 1017:          jsr  ZOUT
5706 4ca457 1018:          jmp  RESALL
5709      1019: ;
5709 8d1557 1020: ZOUT:      sta  ZTEMP+1
570c ad4703 1021:          lda  $0347
570f 48     1022:          PHA
5710 ad4603 1023:          lda  $0346
5713 48     1024:          PHA
5714 a900   1025: ZTEMP:     lda  #0
5716 a200   1026:          ldx  #0           ;1 force NO
5718 60     1027:          RTS
5719      1028: ;
5719      1029: ;
5719      1030: ;   Print text
5719      1031: ;   -----
5719      1032: ; out:
5719      1033: ;   all registers preserved
5719      1034: ; notes:
5719      1035: ;   This print routine will print all characters
5719      1036: ;   following the jsr  PRINT until a delimiter of
5719      1037: ;   -1 ($FF) is reached. PRINT will return to the
5719      1038: ;   point one byte beyond the delimiter.
5719      1039: ;
5719 207457 1040: printsi:  jsr  SAVER
571c ba     1041:          TSX
571d bd0501 1042:          lda  $0105,X
5720 8d2c57 1043:          sta  ZOCH+1
5723 bd0601 1044:          lda  $0106,X
5726 8d2d57 1045:          sta  ZOCH+2
5729      1046: ;
5729 a001   1047:          ldy  #1
572b b9ffff 1048: ZOCH:      lda  $FFFF,Y
572e c9ff   1049:          cmp  #$FF
5730 f006   1050:          beq  ZECHO
5732 200057 1051:          jsr  ECHO
5735 c8     1052:          INY
5736 d0f3   1053:          bne  ZOCH
5738 98     1054: ZECHO:     TYA
5739 18     1055:          CLC
573a 7d0501 1056:          adc  $0105,X
573d 9d0501 1057:          sta  $0105,X
5740 bd0601 1058:          lda  $0106,X
5743 6900   1059:          adc  #0
5745 9d0601 1060:          sta  $0106,X

```

```

5748 4ca457 1061:      jmp  RESALL
574b      1062: ;
574b      1063: ;
574b      1064: ;
574b      1065: ;   Print byte
574b      1066: ;   -----
574b      1067: ; in:
574b      1068: ;   A   = byte to print
574b      1069: ; out:
574b      1070: ;   all registers preserved
574b      1071: ;
574b 207457 1072: ECHO_HEX: jsr  SAVER
574e 205457 1073:      jsr  ZOUTBYT
5751 4ca457 1074:      jmp  RESALL
5754      1075: ;
5754 48      1076: ZOUTBYT:  PHA
5755 4a      1077:      LSR
5756 4a      1078:      LSR
5757 4a      1079:      LSR
5758 4a      1080:      LSR
5759 205d57 1081:      jsr  ZOB
575c 68      1082:      PLA
575d 290f      1083: ZOB:      and  #$0F
575f 0930      1084:      ora  #$30
5761 c93a      1085:      cmp  #$3A
5763 9002      1086:      bcc  ZKKKK
5765 6906      1087:      adc  #6
5767 200057 1088: ZKKKK:      jsr  ECHO
576a 60      1089:      RTS
576b      1090: ;
576b      1091: ;
576b      1092: ;
576b      1093: ;   Print low nibble
576b      1094: ;   -----
576b      1095: ; in:
576b      1096: ;   A   = nibble to print
576b      1097: ; out:
576b      1098: ;   all registers preserved
576b      1099: ;
576b 207457 1100: PRNIB:      jsr  SAVER
576e 205d57 1101:      jsr  ZOB
5771 4ca457 1102:      jmp  RESALL
5774      1103: ;
5774      1104: ;
5774      1105: ;
5774      1106: ;   SAVE & RESTORE registers
5774      1107: ;   -----
5774      1108: ;
5774 08      1109: SAVER:      PHP
5775 48      1110:      PHA
5776 48      1111:      PHA
5777 48      1112:      PHA
5778 08      1113:      PHP

```

```

5779 48 1114:      PHA
577a 8a 1115:      TXA
577b 48 1116:      PHA
577c ba 1117:      TSX
577d bd0901 1118:   lda  $0109,X
5780 9d0501 1119:   sta  $0105,X
5783 bd0701 1120:   lda  $0107,X
5786 9d0901 1121:   sta  $0109,X
5789 bd0101 1122:   lda  $0101,X
578c 9d0701 1123:   sta  $0107,X
578f bd0801 1124:   lda  $0108,X
5792 9d0401 1125:   sta  $0104,X
5795 bd0601 1126:   lda  $0106,X
5798 9d0801 1127:   sta  $0108,X
579b 98 1128:      TYA
579c 9d0601 1129:   sta  $0106,X
579f 68 1130:      PLA
57a0 aa 1131:      TAX
57a1 68 1132:      PLA
57a2 28 1133:      PLP
57a3 60 1134:      RTS
57a4      1135: ;
57a4 68 1136: RESALL:      PLA
57a5 a8 1137:      TAY
57a6 68 1138:      PLA
57a7 aa 1139:      TAX
57a8 68 1140:      PLA
57a9 28 1141:      PLP
57aa 60 1142:      RTS
57ab      1143: ;
57ab      1144: ;
57ab      1145: ;
57ab      1146: ;   Get character
57ab      1147: ;   -----
57ab      1148: ; out:
57ab      1149: ;   A   = character (all others preserved)
57ab      1150: ;
57ab 20b157 1151: GET_KEY:   jsr  ZGETCH
57ae a900 1152: ZCH:      lda  #0
57b0 60 1153:      RTS
57b1      1154: ;
57b1 207457 1155: ZGETCH:      jsr  SAVER
57b4 20bd57 1156:      jsr  ZPHG
57b7 8daf57 1157:      sta  ZCH+1
57ba 4ca457 1158:      jmp  RESALL
57bd      1159: ;
57bd ad25e4 1160: ZPHG:      lda  $E425
57c0 48 1161:      PHA
57c1 ad24e4 1162:      lda  $E424
57c4 48 1163:      PHA
57c5 60 1164:      RTS
57c6      1165: ;
57c6      1166: ;

```

```

57c6      1167: ;
57c6      1168: ;   Get line
57c6      1169: ;   -----
57c6      1170: ; in:
57c6      1171: ;   XA   = buffer address
57c6      1172: ;   Y    = maximum number of characters
57c6      1173: ; out:
57c6      1174: ;   @XA   = data, return ends input, a backspace wor
57c6      1175: ;   all registers preserved
57c6      1176: ; notes:
57c6      1177: ;   This routine will first clear the input window
57c6      1178: ;
57c6 207457 1179: GETLINE:   jsr   SAVER
57c9 8c1058 1180:             sty   ZCNT
57cc 8d0d58 1181:             sta   ZSTOR+1
57cf 8e0e58 1182:             stx   ZSTOR+2
57d2      1183: ;
57d2 a000 1184:             ldy   #0
57d4 a920 1185:             lda   #$20
57d6 200c58 1186: ZCLX:       jsr   ZSTOR
57d9 c8 1187:             INY
57da cc1058 1188:             cpy   ZCNT
57dd d0f7 1189:             bne   ZCLX
57df      1190: ;
57df a000 1191:             ldy   #0
57e1 20ab57 1192: ZINLP:      jsr   GET_KEY
57e4 c97e 1193:             cmp   #$7E
57e6 f012 1194:             beq   ZBS
57e8 c99b 1195:             cmp   #$9B
57ea f01d 1196:             beq   ZENDZ
57ec cc1058 1197:             cpy   ZCNT
57ef f0f0 1198:             beq   ZINLP
57f1 200057 1199:             jsr   ECHO
57f4 200c58 1200:             jsr   ZSTOR
57f7 c8 1201:             INY
57f8 d0e7 1202:             bne   ZINLP
57fa      1203: ;
57fa c000 1204: ZBS:        cpy   #0
57fc f0e3 1205:             beq   ZINLP
57fe 200057 1206:             jsr   ECHO
5801 88 1207:             DEY
5802 a920 1208:             lda   #$20
5804 200c58 1209:             jsr   ZSTOR
5807 d0d8 1210:             bne   ZINLP
5809 4ca457 1211: ZENDZ:      jmp   RESALL
580c      1212: ;
580c 99ffff 1213: ZSTOR:      sta   $FFFF,Y
580f 60 1214:             RTS
5810 00 1215: ZCNT:       dc.b  0
5811      1216: ;
5811      1217: ;
5811      1218: ;=====
5811      1219: ;   write/calculate diskette data

```



```

5811      1220: ;=====
5811      1221: ;   Write all data to disk
5811      1222: ;   -----
5811 a201 1223: WRITE:   ldx  #1                ; W
5813 8e0a03 1224:       stx  SECTOR
5816 ca    1225:       DEX
5817 8e0b03 1226:       stx  SECTOR+1          ; b
581a      1227: ;
581a a240 1228:       ldx  # high BOOTI
581c a000 1229:       ldy  # low BOOTI
581e a903 1230:       lda  #3
5820 20b458 1231:      jsr  WRBLOCK           ; w
5823      1232: ;
5823 ac0f40 1233:      ldy  BOOTI+BMD+SDQBM    ; #
5826 8c025b 1234:      sty  TEMPQX
5829      1235: ;
5829 ad015b 1236:      lda  DATASEC           ; Z
582c 4a    1237:      LSR
582d 4a    1238:      LSR
582e 4a    1239:      LSR
582f aa    1240:      TAX
5830 8e035b 1241:      stx  TEMPQX+1
5833 ad015b 1242:      lda  DATASEC
5836 2907 1243:      and  #7
5838 a8    1244:      TAY
5839 b90e59 1245:      lda  MASK,Y
583c 49ff 1246:      EOR  #$FF
583e      1247: ;
583e 9d095c 1248: CLFIR:   sta  BITMAP,X
5841 a900 1249:      lda  #0
5843 ca    1250:      DEX
5844 10f8 1251:      bpl  CLFIR
5846      1252: ;
5846 ae035b 1253:      ldx  TEMPQX+1
5849 e8    1254:      INX
584a 2c    1255:      dc.b  $2C
584b      1256: ;
584b a200 1257: NXB:   ldx  #0                ; c
584d a9ff 1258:      lda  #$FF
584f 9d095c 1259: SEA:   sta  BITMAP,X
5852 e8    1260:      INX
5853 d0fa 1261:      bne  SEA
5855      1262: ;
5855 ac025b 1263:      ldy  TEMPQX            ; i
5858 88    1264:      DEY
5859 d011 1265:      bne  WRBITX
585b      1266: ;
585b ae055b 1267:      ldx  NIBOFF            ; o
585e bd0e59 1268:      lda  MASK,X
5861 ae045b 1269:      ldx  BYTOFF
5864 9d095c 1270: CLREST:   sta  BITMAP,X
5867 a900 1271:      lda  #0
5869 e8    1272:      INX

```

```

586a d0f8 1273:      bne  CLREST
586c      1274: ;
586c a901 1275: WRBITX:      lda  #1
586e a25c 1276:      ldx  # high BITMAP
5870 a009 1277:      ldy  # low BITMAP
5872 20b458 1278:      jsr  WRBLOCK
5875 ce025b 1279:      dec  TEMPQX
5878 d0d1 1280:      bne  NXB
587a      1281: ;
587a a900 1282:      lda  #0
587c aa 1283:      TAX
587d 9d095d 1284: CLMAPS:      sta  SCMAP,X
5880 9d095e 1285:      sta  DATSEC,X
5883 e8 1286:      INX
5884 d0f7 1287:      bne  CLMAPS
5886      1288: ;
5886 ae0940 1289:      ldx  BOOTI+MDD          ; S
5889 e8 1290:      INX
588a 8e0d5d 1291:      stx  SCMAP+4          ; P
588d a901 1292:      lda  #1
588f a25d 1293:      ldx  # high SCMAP      ; W
5891 a009 1294:      ldy  # low SCMAP
5893 20b458 1295:      jsr  WRBLOCK
5896      1296: ;
5896 a20a 1297:      ldx  #11-1          ; P
5898 bd1759 1298: MVNAM:      lda  NAMTAB,X
589b 9d0f5e 1299:      sta  DATSEC+DEFNAM,X
589e ca 1300:      DEX
589f 10f7 1301:      bpl  MVNAM
58a1 a928 1302:      lda  #DESSUB+DESUSE
58a3 8d095e 1303:      sta  DATSEC+DESTA
58a6 a917 1304:      lda  #DDEX
58a8 8d0c5e 1305:      sta  DATSEC+DENBYT
58ab a901 1306:      lda  #1
58ad a25e 1307:      ldx  # high DATSEC
58af a009 1308:      ldy  # low DATSEC
58b1 4cb458 1309:      jmp  WRBLOCK          ; W
58b4      1310: ;
58b4      1311: ;
58b4      1312: ;   Write a block of sectors
58b4      1313: ;   -----
58b4      1314: ;
58b4 8e0503 1315: WRBLOCK:      stx  DBUFHI          ; a
58b7 8d065b 1316:      sta  NUMSECS          ; n
58ba 8c0403 1317:      sty  DBUFLO
58bd      1318: ;
58bd ad3c5a 1319:      lda  DENSITY          ; g
58c0 ae0b03 1320:      ldx  SECTOR+1
58c3 d009 1321:      bne  ISTDH
58c5 ae0a03 1322:      ldx  SECTOR
58c8 e004 1323:      cpx  #4
58ca b002 1324:      bcs  ISTDH
58cc a980 1325:      lda  #$80

```

```

58ce 8d0803 1326: ISTHD:      sta  DBYTLO
58d1 a200 1327:      ldx  #0
58d3 a8 1328:      TAY
58d4 d002 1329:      bne  ISSNG
58d6 a201 1330:      ldx  #1
58d8 8e0903 1331: ISSNG:      stx  DBYTHI
58db      1332: ;
58db a950 1333: REPOUT:      lda  #'P'
58dd 8d0203 1334:      sta  DCOMND
58e0 a980 1335:      lda  #$80
58e2 8d0303 1336:      sta  DSTATS
58e5 201550 1337:      jsr  XSIO
58e8 1003 1338:      bpl  OKWRTX
58ea 4c6550 1339:      jmp  ERROR
58ed      1340: ;
58ed ee0a03 1341: OKWRTX:      inc  SECTOR
58f0 d003 1342:      bne  SK1
58f2 ee0b03 1343:      inc  SECTOR+1
58f5      1344: ;
58f5 18 1345: SK1:      CLC
58f6 ad0403 1346:      lda  DBUFLO
58f9 6d0803 1347:      adc  DBYTLO
58fc 8d0403 1348:      sta  DBUFLO
58ff ad0503 1349:      lda  DBUFHI
5902 6d0903 1350:      adc  DBYTHI
5905 8d0503 1351:      sta  DBUFHI
5908      1352: ;
5908 ce065b 1353:      dec  NUMSECS
590b d0ce 1354:      bne  REPOUT
590d 60 1355:      RTS
590e      1356: ;
590e      1357: ;
590e 0080c0 1358: MASK:      dc.b  $00,$80,$C0,$E0
5912 f0f8fc 1359:      dc.b  $F0,$F8,$FC,$FE,$FF
5917      1360: ;
5917 4d4149 1361: NAMTAB:      dc.b  "MAIN  "
5922      1362: ;
5922      1363: ;
5922      1364: ;      Calculate all sector positions for given density
5922      1365: ;      -----
5922      1366: ;
5922 ad3e5a 1367: RECALC:      lda  MAXSEC
5925 8d025b 1368:      sta  TEMPQX
5928 ad3f5a 1369:      lda  MAXSEC+1
592b 8d035b 1370:      sta  TEMPQX+1
592e      1371: ;
592e a900 1372:      lda  #0
5930 a203 1373:      ldx  #3      ; d
5932 4e035b 1374: REPROL:      LSR  tempqx+1
5935 6e025b 1375:      ror  TEMPQX
5938 6a 1376:      ROR
5939 ca 1377:      DEX
593a d0f6 1378:      bne  REPROL

```

```

593c 4a 1379:      LSR
593d 4a 1380:      LSR
593e 4a 1381:      LSR
593f 4a 1382:      LSR
5940 4a 1383:      LSR
5941 8d055b 1384:   sta  NIBOFF          ; O
5944      1385: ;
5944 ad025b 1386:   lda  TEMPQX
5947 ae035b 1387:   ldx  TEMPQX+1          ; G
594a      1388: ;
594a 2c3c5a 1389:   bit  DENSITY
594d 1007 1390:   bpl  G2
594f 0a 1391:      ASL
5950 48 1392:      PHA
5951 8a 1393:      TXA
5952 2a 1394:      ROL
5953 aa 1395:      TAX
5954 68 1396:      PLA
5955 4a 1397:      LSR
5956      1398: ;
5956 8d045b 1399: G2:   sta  BYTOFF
5959 e8 1400:      INX
595a 8e0f40 1401:   stx  BOOTI+BMD+SDQBM      ; #
595d 8a 1402:      TXA
595e 18 1403:      CLC
595f 6904 1404:   adc  #4          ; A
5961 8d0940 1405:   sta  BOOTI+MDD          ;
5964 6902 1406:   adc  #2          ; G
5966 8d015b 1407:   sta  DATASEC
5969 8d1440 1408:   sta  BOOTI+BMD+SDDIR      ; F
596c 691e 1409:   adc  #30          ; 1
596e 8d1240 1410:   sta  BOOTI+BMD+SDALLOC      ; F
5971 2c475a 1411:   bit  CURP          ; c
5974 1002 1412:   bpl  ISSOMD
5976 a900 1413:   lda  #0          ; z
5978 29f8 1414: ISSOMD:   and  #$F8
597a      1415: ;
597a 8d2840 1416:   sta  BOOTI+EXTR+DDMAP      ; S
597d      1417: ;
597d 38 1418:      SEC
597e ad3e5a 1419:   lda  MAXSEC          ; #
5981 8d0b40 1420:   sta  BOOTI+BMD+SDTOT
5984 ed015b 1421:   sbc  DATASEC
5987 8d0d40 1422:   sta  BOOTI+BMD+SDFRE
598a ad3f5a 1423:   lda  MAXSEC+1
598d 8d0c40 1424:   sta  BOOTI+BMD+SDTOT+1
5990 e900 1425:   sbc  #0
5992 8d0e40 1426:   sta  BOOTI+BMD+SDFRE+1
5995 ad3c5a 1427:   lda  DENSITY          ; s
5998 8d1f40 1428:   sta  BOOTI+DPD+SPDEN
599b ad425a 1429:   lda  TRACKS
599e 8d1e40 1430:   sta  BOOTI+DPD+SPTRK
59a1      1431: ;

```

```

59a1 ad0ad2 1432:      lda  $D20A          ; g
59a4 8d2740 1433:      sta  BOOTI+EXTR+DRAND
59a7 60 1434:          RTS
59a8 1435: ;
59a8 1436: ;
59a8 1437: ;
59a8 1438: ;=====
59a8 1439: ; data storage
59a8 1440: ;=====
59a8 1441: ;
59a8 44313a 1442: FSPEC:      dc.b  "D1:*"
59ac 2e444f 1443: EXTN:      dc.b  ".DOS", $9B
59b1 1444: ;
59b1 d002d0 1445: MAXST:      dc.w  40*18,40*18,40*26
59b7 d207d2 1446:      dc.w  77*26,77*26,0
59bd a005a0 1447:      dc.w  80*18,80*18,80*26
59c3 760276 1448:      dc.w  35*18,35*18,35*26
59c9 a005a0 1449:      dc.w  40*18*2,40*18*2,40*26*2
59cf a40fa4 1450:      dc.w  77*26*2,77*26*2,0
59d5 400b40 1451:      dc.w  80*18*2,80*18*2,80*26*2
59db ec04ec 1452:      dc.w  35*18*2,35*18*2,35*26*2
59e1 1453: ;
59e1 1454: ; added for ramdisks
59e1 1455: ;      dc.w  32*128,32*128,32*128
59e1 1456: ;
59e1 8000 1457: LEN8T:      dc.b  128,256
59e3 0206 1458: DEN8T:      dc.b  2,6
59e5 1a1a 1459: SEC8T:      dc.b  26,26
59e7 1460: ;
59e7 800080 1461: LEN5T:      dc.b  128,256,128
59ea 000404 1462: DEN5T:      dc.b  0,4,4
59ed 12121a 1463: SEC5T:      dc.b  18,18,26
59f0 1464: ;
59f0 284d50 1465: TRKST:      dc.b  40,77,80,35
59f4 a8cdd0 1466:      dc.b  40+128,77+128,80+128,35+128
59f8 1467: ;      dc.b  32
59f8 1468: ;
59f8 fe5910 1469: ORDT:      dc.w  SD18,DD18,DD26
59fe 04080c 1470: SD18:      dc.b  $04,$08,$0C,$10,$01,$05,$09
5a07 02060a 1471:      dc.b  $02,$06,$0A,$0E,$12,$03,$07
5a10 1472: ;
5a10 010e09 1473: DD18:      dc.b  $01,$0E,$09,$04,$11,$0C,$07
5a19 0a0512 1474:      dc.b  $0A,$05,$12,$0D,$08,$03,$10
5a22 1475: ;
5a22 04080c 1476: DD26:      dc.b  $04,$08,$0C,$10,$14,$18,$01
5a2b 0d1115 1477:      dc.b  $0D,$11,$15,$19,$02,$06,$0A
5a34 161a03 1478:      dc.b  $16,$1A,$03,$07,$0B,$0F,$13
5a3c 1479: ;
5a3c 1480: ;
5a3c 1481: ;
5a3c 1482: DENSITY:      ds.b  1
5a3d 1483: SECTORS:      ds.b  1
5a3e 1484: MAXSEC:      ds.b  2

```

```

5a40 1485: DENS:          ds.b  1
5a41 1486: ;
5a41 1487: INCH8:          ds.b  1
5a42 1488: TRACKS:          ds.b  1
5a43 1489: TRKINX:          ds.b  1
5a44 1490: DENINX:          ds.b  1
5a45 1491: ;
5a45 1492: COUNT:          ds.b  1
5a46 1493: STACKP:          ds.b  1
5a47 1494: CURP:          ds.b  1
5a48 1495: DOSP:          ds.b  1
5a49 1496: DOSPT:          ds.b 10
5a53 1497: DIRP:          ds.b  1
5a54 1498: NAMES:          ds.b 130
5ad6 1499: ENTRY:          ds.b  25
5aef 1500: DOSI:          ds.b  3+8+5
5aff 1501: US:          ds.b  1
5b00 1502: HIGHSP:          ds.b  1
5b01 1503: ;
5b01 1504: DATASEC:          ds.b  1
5b02 1505: TEMPQX:          ds.b  2
5b04 1506: BYTOFF:          ds.b  1
5b05 1507: NIBOFF:          ds.b  1
5b06 1508: NUMSECS:          ds.b  1
5b07 1509: TABLEN:          ds.b  2
5b09 1510: ;
5b09 1511: FMTBUF:          ds.b 256
5c09 1512: BITMAP:          ds.b 256
5d09 1513: SCMAP:          ds.b 256
5e09 1514: DATSEC:          ds.b 256
5f09 1515: TABLE:          ds.b 256
6009 1516: ;
6009 1517: ;
6009 1518: ;~~~~~
6009 1519: ; Ok the relocation marker is to tell the DOS we are now u
6009 1520: ;more data at memlow..
6009 1521: ;
6009 1522: ;
6009 00 1523: initz:          dc.b  0
600a 1524:
600a 1525:
600a 00 1526: zend:          dc.b  0
600b 1527:
600b 1528: ;
600b 1529: ;~~~~~
600b 1530: ;
600b 1531: ;          *** END OF HANDLER ***
600b 1532: ;~~~~~
600b 1533: ;
600b 1534: ;
600b 1535: ; Relocater: main entry
600b 1536: ; -----
600b 1537: ; in:

```

```

600b 1538: ; segtab = table of segment descriptors
600b 1539: ; +00 = relocater table address
600b 1540: ; +02 = originate address of block
600b 1541: ; +04 = destination originate of block
600b 1542: ; +06 = address of block
600b 1543: ; +08 = number of bytes in block
600b 1544: ; +10 = destination address of block
600b 1545: ; +12 = Length of segment descriptor
600b 1546: ; +14 = this is the word location adjust mem
600b 1547:
600b 1548: ;
600b 1549: ;
600b 1550: ;
600b 1551: ;
600b 1552: ; relob = relocater table
600b 1553: ; list of address of words to adjust 2
600b 1554: ; list of address low bytes to adjust 2
600b 1555: ; list of address high bytes to adjust 3
600b 1556: ; followed by their low bytes
600b 1557: ;~~~~~
600b 1558: rlocate:
600b 1559:
600b ad3e61 1560: lda segtab+segaddress ; s
600e 8d3461 1561: sta segtab+dorgadr ; P
6011 8d3a61 1562: sta segtab+blkdes ; P
6014 ad3f61 1563: lda segtab+segaddress+1 ; s
6017 8d3561 1564: sta segtab+dorgadr+1 ; P
601a 8d3b61 1565: sta segtab+blkdes+1 ; P
601d 1566: ; jsr rlocate ; o
601d 1567:
601d 1568:
601d 1569:
601d a900 1570: lda #0 ; s
601f 8dea60 1571: sta segment
6022 1572:
6022 aeea60 1573: segloop: ldx segment
6025 bd3061 1574: lda segtab+rettad,x ; g
6028 8dd060 1575: sta relget+1
602b bd3161 1576: lda segtab+rettad+1,x
602e 8dd160 1577: sta relget+2
6031 0dd060 1578: ora relget+1 ; i
6034 d001 1579: bne havseg
6036 60 1580: rts
6037 1581:
6037 38 1582: havseg: sec
6038 bd3461 1583: lda segtab+dorgadr,x
603b fd3261 1584: sbc segtab+orgadr,x
603e 8deb60 1585: sta zoffset
6041 bd3561 1586: lda segtab+dorgadr+1,x
6044 fd3361 1587: sbc segtab+orgadr+1,x
6047 8dec60 1588: sta zoffset+1
604a 1589:
604a 20db60 1590: zwordlp: jsr getzwp

```

```

604d f013 1591:      beq  zlbytlp          ; i
604f      1592:
604f b1d7 1593:      lda  (zwptr),y          ; a
6051 18   1594:      clc
6052 6deb60 1595:      adc  zoffset
6055 91d7 1596:      sta  (zwptr),y
6057 c8   1597:      iny
6058 b1d7 1598:      lda  (zwptr),y
605a 6dec60 1599:      adc  zoffset+1
605d 91d7 1600:      sta  (zwptr),y
605f 4c4a60 1601:      jmp  zwordlp
6062      1602:
6062 20db60 1603: zlbytlp:  jsr  getzwp          ; g
6065 f00b 1604:      beq  zhbytlp          ; i
6067      1605:
6067 b1d7 1606:      lda  (zwptr),y
6069 18   1607:      clc
606a 6deb60 1608:      adc  zoffset
606d 91d7 1609:      sta  (zwptr),y
606f 4c6260 1610:      jmp  zlbytlp
6072      1611:
6072 20db60 1612: zhbytlp:  jsr  getzwp          ; g
6075 f012 1613:      beq  zmovlp          ; j
6077      1614:
6077 20cf60 1615:      jsr  relget          ; g
607a 18   1616:      clc
607b 6deb60 1617:      adc  zoffset
607e b1d7 1618:      lda  (zwptr),y
6080 6dec60 1619:      adc  zoffset+1
6083 91d7 1620:      sta  (zwptr),y
6085 4c7260 1621:      jmp  zhbytlp
6088 60   1622:      rts
6089      1623:
6089      1624: zmovlp:
6089      1625:
6089 adf960 1626:      lda  segment+segmove          ; g
608c c9ff 1627:      cmp  #seg_off          ; $
608e d001 1628:      bne  .zmovlp          ; m
6090 60   1629:      rts          ; o
6091 aeea60 1630: .zmovlp:  ldx  segment          ; g
6094 bd3661 1631:      lda  segtab+blkadr,x
6097 8db260 1632:      sta  zmovfr+1
609a bd3761 1633:      lda  segtab+blkadr+1,x
609d 8db360 1634:      sta  zmovfr+2
60a0      1635:
60a0 bd3a61 1636:      lda  segtab+blkdes,x
60a3 8db560 1637:      sta  zmovto+1
60a6 bd3b61 1638:      lda  segtab+blkdes+1,x
60a9 8db660 1639:      sta  zmovto+2
60ac      1640:
60ac      1641:          ; g
60ac bc3961 1642:      ldy  segtab+blkbyt+1,x
60af      1643:

```



```

60af a200 1644:      ldx  #0
60b1 bdffff 1645: zmovfr:      lda  $ffff,x
60b4 9dffff 1646: zmovto:      sta  $ffff,x
60b7 e8      1647:      inx
60b8 d0f7 1648:      bne  zmovfr
60ba eeb360 1649:      inc  zmovfr+2
60bd eeb660 1650:      inc  zmovto+2
60c0 88      1651:      dey
60c1 10ee 1652:      bpl  zmovfr          ; m
60c3      1653:                      ;
60c3 adea60 1654:      lda  segment
60c6 18      1655:      clc
60c7 690c 1656:      adc  #seglen        ; g
60c9 8dea60 1657:      sta  segment
60cc 4c2260 1658:      jmp  segloop
60cf      1659:
60cf      1660:
60cf adffff 1661: relget:      lda  $ffff
60d2 eed060 1662:      inc  relget+1
60d5 d003 1663:      bne  nc1
60d7 eed160 1664:      inc  relget+2
60da 60      1665: nc1:      rts
60db      1666:
60db 20cf60 1667: getzwp:      jsr  relget
60de 85d7 1668:      sta  zwptr          ; s
60e0 20cf60 1669:      jsr  relget
60e3 a000 1670:      ldy  #0
60e5 85d8 1671:      sta  zwptr+1
60e7 05d7 1672:      ora  zwptr          ; c
60e9 60      1673:      rts
60ea      1674:
60ea 00      1675: segment:      dc.b  0          ; c
60eb 0000 1676: zoffset:      dc.w  0          ; o
60ed      1677: ;
60ed      1678: ;~~~~~
60ed      1679: ;
60ed      1680: ;      Relocation data table
60ed      1681: ;      -----
60ed      1682: ;      We resolve Word locations First!
60ed      1683: ;
60ed      1684: ;      Things like  lda  $5000
60ed      1685: ;                  sta  $5000
60ed      1686: ;                  lda  $5000,y
60ed      1687: ;                  jsr  $5000
60ed      1688: ;                  jmp  $5000
60ed      1689: ;
60ed      1690: ;      Well you get the Idea..
60ed      1691: ;
60ed      1692: ;~~~~~
60ed      1693: ;
60ed      1694: ; $3000 tabel
60ed      1695: ;
60ed      1696: rtable1:

```

```

60ed 0740 1697:      dc.w  w01+1
60ef 3f40 1698:      dc.w  w02+1
60f1 4540 1699:      dc.w  w03+1
60f3 4f40 1700:      dc.w  w04+1
60f5 5540 1701:      dc.w  w05+1
60f7 5b40 1702:      dc.w  w06+1
60f9 6440 1703:      dc.w  w07+1
60fb 8340 1704:      dc.w  w08+1
60fd 9340 1705:      dc.w  w09+1
60ff 9840 1706:      dc.w  w10+1
6101      1707:
6101 ab40 1708:      dc.w  w11+1
6103 b040 1709:      dc.w  w12+1
6105 b940 1710:      dc.w  w13+1
6107 c140 1711:      dc.w  w14+1
6109 cc40 1712:      dc.w  w15+1
610b cf40 1713:      dc.w  w16+1
610d d240 1714:      dc.w  w17+1
610f ef40 1715:      dc.w  w18+1
6111 1341 1716:      dc.w  w19+1
6113 1e41 1717:      dc.w  w20+1
6115      1718:
6115 2541 1719:      dc.w  w21+1
6117 3241 1720:      dc.w  w22+1
6119 4041 1721:      dc.w  w23+1
611b 6141 1722:      dc.w  w24+1
611d 6c41 1723:      dc.w  w25+1
611f      1724:
611f      1725:
611f 0000 1726:      dc.w  0              ; o
6121      1727: ;
6121      1728: ;
6121      1729: ;Ok these are low Byte Locations.. only one byte changed!
6121      1730: ; OK Resolving Low Byte's First
6121      1731: ;
6121 a140 1732:      dc.w  lb_1+1
6123 d840 1733:      dc.w  lb_2+1
6125 0000 1734:      dc.w  0              ; o
6127      1735: ;~~~~~
6127      1736: ;Now to resolve High Bytes..... Only one byte to change!
6127      1737: ; Here is what is needed... location of high byte in .wor
6127      1738: ; Second thing needed is the low byte of the address you a
6127      1739: ;to adjust!
6127      1740: ;
6127 a640 1741:      dc.w  hb_1+1
6129 78   1742:      dc.b  LRTS
612a da40 1743:      dc.w  hb_2+1
612c 30   1744:      dc.b  NODMSG
612d      1745:
612d 0000 1746:      dc.w  0              ; o
612f 00   1747:      dc.b  0              ; o
6130      1748:
6130      1749:

```

```
6130      1750:
6130      1751: ;~~~~~
6130      1752: ;End of relocation Table!!!
6130      1753: ;master table:
6130      1754: ;
6130      1755: segtab:
6130 ed60 1756:      dc.w  rtable1      ; a
6132 0040 1757:      dc.w  initz1        ; a
6134 0000 1758:      dc.w  0              ; r
6136 0040 1759:      dc.w  initz1        ; a
6138 8001 1760:      dc.w  zend1-initz1    ; s
613a 0000 1761:      dc.w  0              ; d
613c 0000 1762:      dc.w  0              ; E
613e 0030 1763:      dc.w  $3000          ;se
6140 ff   1764:      dc.b  seg_off      ; -
6141      1765:
6141      1766:
6141      1767: win_end:    ds.b  0
6141      1768: ;
```

End assembly: no errors