


```

0000 1: ;-----
0000 2: ;
0000 3: ;      Copyright 2012 Intergraded Logic Systems
0000 4: ;      Source Code is Copyright Stephen
0000 5: ;
0000 6: ;
0000 7: ;      Please do not share this source!
0000 8: ;-----
0000 9: ;      Xtype.com
0000 10: ;      This program is to replace
0000 11: ;      command line Type command
0000 12: ;
0000 13: ;Added stuff for walter.
0000 14: ; found a bug and fixed it.. if the first parm is not lega
0000 15: ;
0000 16: ;added tab conversion for 127==>$09  $09==>127
0000 17: ;
0000 18: ;
0000 19: ;      ~~~~~
0000 20: ;-----
0000 21: ; Notes: o This source code MAY NOT be placed for download
0000 22: ;      o "mea" is a macro that loads the address of the
0000 23: ;      into the pointer specified by the second fi
0000 24: ;-----
0000 25: ; Assembler: MADMAC (tm) ST Cross Assembler (Atari Corp)
0000 26: ;      XASM st and IBM VERSIONS
0000 27: ;-----
0000 28: com_file_name:      .macro
0000 29: ;      dc.b 12,"filename.ext",$9b
0000 30: ;      dc.b 12,"Xtype.com  ",$9b
0000 31: ;      .endm
0000 32: ;
0000 33: ;
0000 34: ;~~~~~
0000 35: ; File revision history. Version Number in hex
0000 36: ;
0000 37: file_ver:      equ $19
0000 38: ;~~~~~
0000 39: ; the Month in dec for the first time this revision
0000 40: ;was compiled
0000 41: ;
0000 42: c_month:      equ 10
0000 43: ;~~~~~
0000 44: ; the day in dec for the first time this revision
0000 45: ;was compiled
0000 46: c_day:      equ 19
0000 47: ;~~~~~
0000 48: ; the year in dec for the first time this revision
0000 49: ;was compiled
0000 50: c_year:      equ 2012
0000 51: ;~~~~~
0000 52: ; Type of compiler used in program
0000 53: ;

```

```

0000      54: ; 0 = unknown Compiler
0000      55: ; 1 = xasm
0000      56: ; 2 = mac_65
0000      57: ; 3 = basic
0000      58: ; 4 = compiled basic xl
0000      59: ; 5 = C65
0000      60: ; 6 = Action
0000      61: ;
0000      62: ; We can add more as time goes on!
0000      63: ;
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: ;
0000      83: l_relocatable:      equ    r..no
0000      84: ;~~~~~
0000      85: ; Gotta define our Header control Byte
0000      86: ;$ff=bypass intro
0000      87: ;$15 = Bypass Dos check
0000      88: r..crl:             equ    $7d
0000      89: r..crlf:            equ    $9b
0000      90: r..space:           equ    $20
0000      91: r..bypassDOS:       equ    $15
0000      92: r..bypass:          equ    $ff
0000      93: r..supress:         equ    $ff
0000      94:
0000      95:
0000      96: l_frstscreenbyte:   equ    r..bypass
0000      97: ;~~~~~
0000      98: ; The language the output file is in.
0000      99: ;
0000     100: ;
0000     101: ; 0 =   Undefined
0000     102: ; 1 =   English
0000     103: ; 2 =   German
0000     104: ;
0000     105: r..language:        equ    1
0000     106: ;~~~~~

```

```

0000      107:
0000      108:
0000      109:          .include    equates
0000      110:          .include    globals
0000      111:          .include    macros
0000      112:
4000      113:          .org    $4000
4000      114:
4000      115: win_start:
4000      116: header_info:
4000      117:          .include    header
42a7      118: ;
42a7      119: start:
42a7 a210  120:          ldx    #disk1
42a9 20304a 121:          jsr    close
42ac a220  122:          ldx    #disk2
42ae 20304a 123:          jsr    close
42b1      124:
42b1 a50a  125:          lda    comtab
42b3 18    126:          CLC
42b4 6903  127:          adc    # low ZCRNAME
42b6 8da943 128:          sta    cli.crunch+1
42b9 a50b  129:          lda    comtab+1
42bb 6900  130:          adc    # high ZCRNAME
42bd 8daa43 131:          sta    cli.crunch+2
42c0 20a843 132:          jsr    cli.crunch
42c3 d003   133:          bne    .get_parm1
42c5 4cab43 134:          jmp    parm_errors
42c8      135:
42c8      136: .get_parm1:
42c8 a50a  137:          lda    comtab
42ca 18    138:          CLC
42cb 6921  139:          adc    # low COMFNAM
42cd 8582  140:          sta    s_ptr
42cf a50b  141:          lda    comtab+1
42d1 6900  142:          adc    # high COMFNAM
42d3 8583  143:          sta    s_ptr+1
42d5      144:          mea    parm_1,d_ptr
42d5 a9bf   ---:          lda    #low parm_1
42d7 8584   ---:          sta    d_ptr
42d9 a945   ---:          lda    #high parm_1
42db 8585   ---:          sta    d_ptr + 1
42dd 20524c 145:          jsr    tostr
42e0      146:
42e0      147:
42e0 20a843 148:          jsr    cli.crunch
42e3 d003   149:          bne    .get_parm2
42e5 4c3643 150:          jmp    .exit
42e8      151:
42e8 a50a  152: .get_parm2:          lda    comtab
42ea 18    153:          CLC
42eb 6921  154:          adc    # low COMFNAM
42ed 8582  155:          sta    s_ptr

```

```

42ef a50b 156:      lda  comtab+1
42f1 6900 157:      adc  # high COMFNAM
42f3 8583 158:      sta  s_ptr+1
42f5      159:      mea  parm_2,d_ptr
42f5 a90f ----:      lda  #low parm_2
42f7 8584 ----:      sta  d_ptr
42f9 a946 ----:      lda  #high parm_2
42fb 8585 ----:      sta  d_ptr + 1
42fd 20524c 160:      jsr  tostr
4300      161:
4300 a004 162:      ldy  #$04
4302 cc0f46 163:      cpy  parm_2
4305 b02f 164:      bcs  .exit
4307 b90f46 165:      lda  parm_2,y
430a c92f 166:      cmp  #'/'
430c d028 167:      bne  .exit
430e c8 168: .p2:      iny
430f b90f46 169:      lda  parm_2,y
4312 20364c 170:      jsr  upcase
4315 c94e 171:      cmp  #'N'
4317 d003 172:      bne  .pa
4319 ee6b45 173:      inc  need_pause
431c c941 174: .pa:      cmp  #'A'
431e d003 175:      bne  .pb
4320 ee6d45 176:      inc  do_type
4323 c952 177: .pb:      cmp  #'R'
4325 d003 178:      bne  .pc
4327 ee6c45 179:      inc  strip
432a c953 180: .pc:      cmp  #'S'
432c d003 181:      bne  .p.done
432e ee6e45 182:      inc  speed
4331      183: .p.done:
4331 cc0f46 184:      cpy  parm_2
4334 90d8 185:      bcc  .p2
4336      186:
4336      187:
4336      188: .exit:
4336      189:
4336      190:      mea  parm_1,s_ptr
4336 a9bf ----:      lda  #low parm_1
4338 8582 ----:      sta  s_ptr
433a a945 ----:      lda  #high parm_1
433c 8583 ----:      sta  s_ptr + 1
433e      191:      mea  filename,d_ptr
433e a96f ----:      lda  #low filename
4340 8584 ----:      sta  d_ptr
4342 a945 ----:      lda  #high filename
4344 8585 ----:      sta  d_ptr + 1
4346 20924c 192:      jsr  toascii
4349      193:
4349      194:
4349 ad1fd0 195:      lda  consol
434c c907 196:      cmp  #consol_any

```

```

434e f03c 197:          beq  .no_tell
4350      198:
4350 20394a 199:          jsr  printsi
4353 9b2043 200:          dc.b  $9b," CLI Parms"
435e 9b5061 201:          dc.b  $9b,"Parm_1 =",-1
4368      202:          mea  parm_1,pr_ptr
4368 a9bf ----:          lda  #low parm_1
436a 8588 ----:          sta  pr_ptr
436c a945 ----:          lda  #high parm_1
436e 8589 ----:          sta  pr_ptr + 1
4370 20124c 203:          jsr  echos
4373      204:
4373 20394a 205:          jsr  printsi
4376 9b5061 206:          dc.b  $9b,"Parm_2 =",-1
4380      207:          mea  parm_2,pr_ptr
4380 a90f ----:          lda  #low parm_2
4382 8588 ----:          sta  pr_ptr
4384 a946 ----:          lda  #high parm_2
4386 8589 ----:          sta  pr_ptr + 1
4388 20124c 208:          jsr  echos
438b      209:
438b      210:
438b 60      211:          rts
438c      212:
438c adc145 213: .no_tell:          lda  parm_1+2          ;0,
438f c931 214:          cmp  #'1'
4391 301b 215:          bmi  show_help
4393 c93a 216:          cmp  #'9'+1
4395 b017 217:          bcs  show_help
4397 adc245 218:          lda  parm_1+3
439a c93a 219:          cmp  #':'
439c d010 220:          bne  show_help
439e adc345 221:          lda  Parm_1+4
43a1 c92f 222:          cmp  # '/'
43a3 f009 223:          beq  show_help
43a5      224:
43a5 4caf47 225:          jmp  type
43a8      226:
43a8 4c0000 227: cli.crunch:          jmp  $0000
43ab      228:
43ab      229: parm_errors:
43ab 20a442 230:          jsr  start-3
43ae 20394a 231: Show_help:          jsr  printsi
43b1 9b9b 232:          dc.b  $9B,$9B
43b3 20      233:          dc.b  " "
43b4 d3f9ee 234:          dc.b  <+128>,"Syntax"
43ba 20666f 235:          dc.b  " for use: ",$9b
43c4 205874 236:          dc.b  " Xtype "
43cb dbc4ee 237:          dc.b  <+128>,"[Dn:>dir>]"
43d5 666e61 238:          dc.b  "fname.ext [parm]", $9b
43e6 204472 239:          dc.b  " Drive and path ", $9b,$9b
43f8 205061 240:          dc.b  " Parm = /N Non-stop Read",
4411 205061 241:          dc.b  " Parm = /R Remove Inverse"

```

```

442b 205061 242:      dc.b  " Parm = /A Ascii to Atasci
4447 205061 243:      dc.b  " Parm = /S Slow screen dis
4466 205061 244:      dc.b  " Parm = /NRAS in any order
4485 9b9b 245:      dc.b  $9b,$9b
4487 205468 246:      dc.b  " These Functions are "
449c c8eff4 247:      dc.b  <+128>,"Hot Keyed"
44a5 9b 248:      dc.b  $9b
44a6 20416e 249:      dc.b  " And can be called while a
44c5 9b 250:      dc.b  $9b
44c6 206265 251:      dc.b  " being Xtyped!"
44d4 9b 252:      dc.b  $9b
44d5 20 253:      dc.b  " "
44d6 dbd3e5 254:      dc.b  <+128>,"[Select]"
44de 3d2074 255:      dc.b  "= toggles "
44e8 dbadcd 256:      dc.b  <+128>,"[-More-]"
44f0 9b 257:      dc.b  $9b
44f1 20 258:      dc.b  " "
44f2 dbd3f4 259:      dc.b  <+128>,"[Start]"
44f9 203d20 260:      dc.b  " = toggles Inverse strip",
4512 20 261:      dc.b  " "
4513 dbcff0 262:      dc.b  <+128>,"[Option]"
451b 3d2074 263:      dc.b  "= toggles Screen Speed",$9
4532 20 264:      dc.b  " "
4533 dbc2f2 265:      dc.b  <+128>,"[Brk]"
4538 204162 266:      dc.b  " Aborts File",$9b
4545 20 267:      dc.b  " "
4546 dbc3ef 268:      dc.b  <+128>,"[Control+N]"
4551 205175 269:      dc.b  " Quits File operation",$9b
4567 9b 270:      dc.b  $9b
4568 271:
4568 9bff 272:      dc.b  $9b,-1
456a 60 273:      RTS
456b 274:
456b 275: ;~~~~~
456b 276: ; Vars
456b 00 277: need_pause:      dc.b  0
456c 00 278: strip:          dc.b  0
456d 00 279: do_type:        dc.b  0
456e 00 280: speed:          dc.b  0
456f 281: FILENAME:      ds.b  80
45bf 282:
45bf 283: parm_1:         ds.b  80
460f 284: parm_2:         ds.b  80
465f 285: parm_3:         ds.b  80
46af 286: in_str:         ds.b  128
472f 287: instr:          ds.b  128
47af 288: ;
47af 289:
47af 290:
47af 291: ;      View a text file to the screen
47af 292: ;      ~~~~~
47af 293: type:
47af ad6d45 294:      lda  do_type

```

```

47b2 8d5d48 295:          sta  .xlate
47b5      296:
47b5      297:
47b5      298:          mea  FILENAME,fn_ptr
47b5 a96f  ----:          lda  #low FILENAME
47b7 8586  ----:          sta  fn_ptr
47b9 a945  ----:          lda  #high FILENAME
47bb 8587  ----:          sta  fn_ptr + 1
47bd a220  299:          ldx  #disk_2
47bf 20b949 300:          jsr  open_in
47c2 1003  301:          bpl  .clearscr
47c4 4c044a 302:          jmp  system_error      ;da
47c7      303:
47c7 a97d  304: .clearscr:          lda  #125
47c9 20c24b 305:          jsr  putlocal
47cc 20a442 306:          jsr  start-3            ; d
47cf      307:
47cf a220  308: .read_loop:          ldx  #disk_2
47d1 a907  309:          lda  #ccgetch
47d3 9d4203 310:          sta  iccom,x
47d6 a900  311:          lda  #0                ;ge
47d8 9d4403 312:          sta  icbal,x
47db 9d4503 313:          sta  icbah,x
47de 9d4803 314:          sta  icbll,x
47e1 9d4903 315:          sta  icblh,x
47e4 2056e4 316:          jsr  ciov              ;ge
47e7 1010  317:          bpl  .readOK
47e9 98     318: .readerr:          tya
47ea 48     319:          pha
47eb a220  320:          ldx  #disk_2          ;cl
47ed 20304a 321:          jsr  close
47f0 68     322:          pla
47f1 a8     323:          tay
47f2 c088  324:          cpy  #136            ;wa
47f4 f066  325:          beq  .exit           ;ye
47f6 4c044a 326:          jmp  system_error
47f9      327:
47f9      328: .readOK:
47f9 ae5d48 329:          ldx  .xlate          ;tr
47fc f021  330:          beq  .noxlate
47fe c97e  331:          cmp  #'~'
4800 d002  332:          bne  .bks
4802 a92d  333:          lda  #'-'
4804      334:
4804 c97f  335: .bks:          cmp  #127        ; A
4806 d004  336:          bne  .tab
4808 a909  337:          lda  #$09
480a d006  338:          bne  .ascii
480c      339:
480c c909  340: .tab:          cmp  #$09        ;as
480e d002  341:          bne  .ascii
4810 a97f  342:          lda  #127            ;at
4812      343: .ascii:

```



```

4812 c90a 344:      cmp    #10          ;li
4814 f022 345:      beq    .skip_it        ;ye
4816 c90d 346:      cmp    #13
4818 d005 347:      bne    .noxlate
481a a99b 348:      lda    #$9b
481c 4c2a48 349:     jmp    .nostrip
481f      350: .noxlate:
481f c99b 351:      cmp    #$9b
4821 f007 352:      beq    .nostrip
4823 ae6c45 353:     ldx    Strip
4826 f002 354:      beq    .nostrip
4828 297f 355:      and    #$7f
482a      356: .nostrip:
482a ae6e45 357:     ldx    speed
482d f006 358:      beq    .fast
482f 20c24b 359:     jsr    putlocal          ; S
4832 4c3848 360:     jmp    .skip_it
4835      361:
4835 208f4a 362: .fast:      jsr    fast_output      ;Fa
4838      363:
4838      364: .skip_it:
4838 20a348 365:     jsr    check_break
483b ad6b45 366:     lda    need_pause
483e c900 367:      cmp    #$00
4840 d013 368:      bne    .notatbot
4842      369:
4842 a554 370:      lda    row
4844 c917 371:      cmp    #23              ;at
4846 900d 372:      bcc    .notatbot
4848      373:
4848 a900 374:      lda    #$00
484a 8d6b45 375:     sta    need_pause
484d 205f48 376:     jsr    more
4850      377:
4850 a97d 378:      lda    #125
4852 20c24b 379:     jsr    putlocal
4855 4ccf47 380: .notatbot:   jmp    .read_loop
4858 a088 381: .aborted:   ldy    #136              ;fa
485a d08d 382:      bne    .readerr
485c 60 383: .exit:    rts
485d      384:
485d 00 385: .xlate:   dc.b    0
485e 00 386: .temp.a   dc.b    0
485f      387:
485f      388: ;
485f      389: more:
485f a9ff 390:      lda    #$ff
4861 8dfc02 391:     sta    764
4864 20394a 392:     jsr    printsi
4867 dba0ad 393:     dc.b    <+128>,"[ - MORE - ]"
4873 ff 394:      dc.b    -1
4874 20a449 395:     jsr    get_key
4877 c97e 396:      cmp    #$7e              ; b

```

```

4879 f015 397:      beq  .esc_more
487b c91b 398:      cmp  #$1b      ;es
487d f011 399:      beq  .esc_more
487f      400: ;      cmp  #$1c      ;up
487f      401: ;      beq  .esc_more
487f      402:
487f      403:
487f 20394a 404:      jsr  printsi
4882 7e7e7e 405:      dc.b  $7e,$7e,$7e,$7e
4886 7e7e7e 406:      dc.b  $7e,$7e,$7e,$7e
488a 7e7e7e 407:      dc.b  $7e,$7e,$7e,$7e
488e ff 408:      dc.b  -1
488f 60 409:      rts
4890      410:
4890      411: .esc_more:
4890 a900 412:      lda  #$00
4892 8df002 413:      sta  752
4895 a220 414:      ldx  #disk_2      ;cl
4897 20304a 415:      jsr  close
489a 20394a 416:      jsr  printsi
489d 9b9bff 417:      dc.b  $9b,$9b,-1
48a0 68 418:      pla
48a1 68 419:      pla
48a2 60 420:      rts
48a3      421:
48a3      422:
48a3      423:
48a3      424: check_break:
48a3 ad1fd0 425:      lda  consol
48a6 c900 426:      cmp  #0      ; C
48a8 f04a 427:      beq  .quit      ; c
48aa c905 428:      cmp  #5      ; S
48ac f071 429:      beq  ..Select      ; n
48ae c906 430:      cmp  #6      ; S
48b0 f070 431:      beq  ..Start      ; n
48b2 c903 432:      cmp  #3      ; O
48b4 f06f 433:      beq  ..Option      ; n
48b6      434:
48b6      435:
48b6 adfc02 436:      lda  764      ; C
48b9 c5ff 437:      cmp  $ff
48bb f013 438:      beq  .do_brk
48bd c91c 439:      cmp  #$1c      ; e
48bf f033 440:      beq  .quit
48c1 c9a3 441:      cmp  #$a3      ; C
48c3 f02f 442:      beq  .quit      ; g
48c5 c921 443:      cmp  #$21      ; S
48c7 f05f 444:      beq  .do_more      ; N
48c9 c966 445:      cmp  #$66      ; "
48cb d003 446:      bne  .do_brk
48cd 4c2b49 447:      jmp  .help      ; L
48d0      448:
48d0      449:

```

```

48d0 a511 450: .do_brk:      lda   brkkey
48d2 c900 451:                cmp   #$00
48d4 d048 452:                bne   .exit
48d6 a980 453:                lda   #$80
48d8 8511 454:                sta   brkkey
48da a900 455:                lda   #$00
48dc 8df002 456:               sta   752
48df 20394a 457:               jsr   printsi
48e2 9b 458:                dc.b   $9b
48e3 c2f2e5 459:               dc.b   <+128>,"Break"
48e8 204162 460:               dc.b   " Aborted!",$9b
48f2 9bff 461:                dc.b   $9b,-1
48f4      462:
48f4 a220 463: .quit:         ldx   #disk_2      ;cl
48f6 20304a 464:               jsr   close
48f9 a9ff 465:                lda   #$ff
48fb 8dfc02 466:               sta   ch
48fe 20394a 467:               jsr   printsi
4901 9b 468:                dc.b   $9b
4902 205874 469:               dc.b   " Xtype Operation Ended!",$
491a 9bff 470:                dc.b   $9b,-1
491c 68 471:                pla
491d 68 472:                pla
491e 60 473: .exit:        rts
491f      474:
491f 4c5f49 475: ..Select:     jmp   toggle_More
4922 4c7649 476: ..Start:      jmp   toggle_strip
4925 4c8d49 477: ..Option:     jmp   toggle_speed
4928 4c5f48 478: .do_more:     jmp   more
492b      479:
492b 20394a 480: .help:        jsr   printsi
492e 9b9b 481:                dc.b   $9b,$9b
4930 203d3d 482:                dc.b   " =====
4958 20ae43 483:                jsr   Show_help
495b 205f48 484:                jsr   more
495e 60 485:                rts
495f      486:
495f      487:
495f      488:
495f      489:
495f      490: toggle_More:
495f ad1fd0 491: .hold:        lda   consol
4962 c907 492:                cmp   #$07      ; N
4964 d0f9 493:                bne   .hold      ; w
4966 ad6b45 494:                lda   need_pause
4969 c900 495:                cmp   #$00
496b f003 496:                beq   .a
496d a900 497:                lda   #$00
496f 2c 498:                dc.b   $2c      ; t
4970 a901 499: .a:           lda   #$01
4972 8d6b45 500:               sta   need_pause
4975 60 501:                rts
4976      502:

```

```

4976      503: toggle_strip:
4976 ad1fd0 504: .hold:      lda  consol
4979 c907  505:      cmp  #$07          ; N
497b d0f9  506:      bne  .hold          ; w
497d ad6c45 507:      lda  strip
4980 c900  508:      cmp  #$00
4982 f003  509:      beq  .a
4984 a900  510:      lda  #$00
4986 2c    511:      dc.b $2c          ; t
4987 a901  512: .a:      lda  #$01
4989 8d6c45 513:      sta  strip
498c 60    514:      rts
498d      515:
498d      516: toggle_speed:
498d ad1fd0 517: .hold:      lda  consol
4990 c907  518:      cmp  #$07          ; N
4992 d0f9  519:      bne  .hold          ; N
4994 ad6e45 520:      lda  speed
4997 c900  521:      cmp  #$00
4999 f003  522:      beq  .a
499b a900  523:      lda  #$00
499d 2c    524:      dc.b $2c          ; t
499e a901  525: .a:      lda  #$01
49a0 8d6e45 526:      sta  speed
49a3 60    527:      rts
49a4      528:
49a4      529:
49a4      530: ;~~~~~
49a4      531: ; GET KEY ROUTEEN
49a4      532: ;this routeen get's one key and
49a4      533: ;return it to the accumulator
49a4      534: ;~~~~~
49a4      535: ;
49a4 86b6  536: get_key:      stx  $b6
49a6 84b7  537:      sty  $b7
49a8 20b049 538:      jsr  .a
49ab a6b6  539:      ldx  $b6
49ad a4b7  540:      ldy  $b7
49af 60    541:      rts
49b0 ad25e4 542: .a:      lda  $e420+5
49b3 48    543:      pha
49b4 ad24e4 544:      lda  $e420+4
49b7 48    545:      pha
49b8 60    546:      rts
49b9      547:
49b9      548:
49b9      549: ;~~~~~
49b9      550: ; This is my ciov open file Calls
49b9      551: ;~~~~~
49b9      552:
49b9 a904  553: open_in:      lda  #4
49bb 4cd749 554:      jmp  do_open
49be      555:

```

```

49be a908 556: open_out:      lda  #8
49c0 4cd749 557:              jmp  do_open
49c3      558:
49c3 a909 559: open_app:      lda  #9
49c5 4cd749 560:              jmp  do_open
49c8      561:
49c8 a90c 562: open_updt:      lda  #12
49ca 4cd749 563:              jmp  do_open
49cd      564:
49cd a906 565: open_dir:      lda  #6
49cf 4cd749 566:              jmp  do_open
49d2      567:
49d2 a914 568: open_raw:      lda  #20              ;op
49d4 4cd749 569:              jmp  do_open
49d7      570:
49d7      571: do_open
49d7 48 572:          pha              ;op
49d8 8a 573:          txa              ;
49d9 48 574:          pha
49da 20304a 575:          jsr  close        ;ma
49dd 68 576:          pla
49de aa 577:          tax
49df 68 578:          pla
49e0 9d4a03 579:          sta  icaux1,x
49e3 a903 580:          lda  #ccopen
49e5 9d4203 581:          sta  iccom,x
49e8 a586 582:          lda  fn_ptr
49ea 9d4403 583:          sta  icbal,x
49ed a587 584:          lda  fn_ptr+1
49ef 9d4503 585:          sta  icbah,x
49f2 a950 586:          lda  #80
49f4 9d4803 587:          sta  icbll,x
49f7 a900 588:          lda  #0
49f9 9d4903 589:          sta  icblh,x
49fc a980 590:          lda  #128          ;fo
49fe 9d4b03 591:          sta  icaux2,x
4a01 4c56e4 592:          jmp  ciov
4a04      593:
4a04      594:
4a04      595: ;   Display error message of flag in Y register
4a04      596: ;   ~~~~~
4a04      597:
4a04 8c2f4a 598: system_error:      sty  .hold
4a07 a220 599:          ldx  #$20
4a09 20304a 600:          jsr  close
4a0c ee6b45 601:          inc  need_pause    ;pa
4a0f 20394a 602:          jsr  printsi
4a12 9b9b 603:          dc.b  $9b,$9b
4a14 537973 604:          dc.b  "System Error: ",-1
4a23 ad2f4a 605:          lda  .hold
4a26 20d24b 606:          jsr  pr_byte
4a29 20394a 607:          jsr  printsi
4a2c 9bff 608:          dc.b  $9b,-1

```

```

4a2e 60    609:          rts
4a2f      610:
4a2f      611:
4a2f 00    612: .hold:          dc.b  0
4a30      613:
4a30      614:
4a30      615: ;~~~~~
4a30      616: ; Genric Close command
4a30      617: ;~~~~~
4a30      618: close:
4a30 a90c  619:          lda  #$0c
4a32 9d4203 620:          sta  $0342,x
4a35 2056e4 621:          jsr  ciov
4a38 60    622:          rts
4a39      623:
4a39      624: ;
4a39      625: ;
4a39      626: ;-----
4a39      627: ;prints Routine macro
4a39      628: ;this is a slow print to the screen
4a39      629: ;usage jsr  prints
4a39      630: ; .byte $9b," string to be printed",$ff
4a39      631: ;-----
4a39      632: ;
4a39      633: prints:
4a39 68    634:          pla
4a3a 8d4a4a 635:          sta  .pstr+1
4a3d 68    636:          pla
4a3e 8d4b4a 637:          sta  .pstr+2
4a41 ee4a4a 638: .prsl:    inc  .pstr+1
4a44 d003   639:          bne  .pstr
4a46 ee4b4a 640:          inc  .pstr+2
4a49 adffff 641: .pstr:    lda  $ffff
4a4c c9ff   642:          cmp  #$ff
4a4e f006   643:          beq  .estri
4a50 208f4a 644:          jsr  fast_output
4a53 4c414a 645:          jmp  .prsl
4a56 ad4b4a 646: .estri:   lda  .pstr+2
4a59 48    647:          pha
4a5a ad4a4a 648:          lda  .pstr+1
4a5d 48    649:          pha
4a5e 60    650:          rts
4a5f      651: ;
4a5f      652: ;
4a5f      653:
4a5f      654: ;
4a5f      655: ;-----
4a5f      656: ;actual screen handler
4a5f      657: ;screen offset definitions
4a5f      658: ;-----
4a5f      659: scr_offset:
4a5f 0000   660:          dc.w  0
4a61 2800   661:          dc.w  40

```

```

4a63 5000 662:          dc.w 80
4a65 7800 663:          dc.w 120
4a67 a000 664:          dc.w 160
4a69 c800 665:          dc.w 200
4a6b f000 666:          dc.w 240
4a6d 1801 667:          dc.w 280
4a6f 4001 668:          dc.w 320
4a71 6801 669:          dc.w 360
4a73 9001 670:          dc.w 400
4a75 b801 671:          dc.w 440
4a77 e001 672:          dc.w 480
4a79 0802 673:          dc.w 520
4a7b 3002 674:          dc.w 560
4a7d 5802 675:          dc.w 600
4a7f 8002 676:          dc.w 640
4a81 a802 677:          dc.w 680
4a83 d002 678:          dc.w 720
4a85 f802 679:          dc.w 760
4a87 2003 680:          dc.w 800
4a89 4803 681:          dc.w 840
4a8b 7003 682:          dc.w 880
4a8d 9803 683:          dc.w 920
4a8f      684: ;
4a8f      685: echo:
4a8f      686: fast_output:
4a8f 855a 687:          sta $5a
4a91 ad0407 688:         lda  sc.redirect
4a94 c902 689:          cmp  #print_p
4a96 f01f 690:         beq  .go_cio
4a98      691: .leo:
4a98 adff02 692:         lda  $02ff      ; CONTROL 1
4a9b c900 693:          cmp  #$00
4a9d d0f9 694:          bne  .leo
4a9f      695:
4a9f a55a 696:          lda  $5a
4aa1 c920 697:          cmp  #32
4aa3 9004 698:          bcc  .do_look
4aa5 c97d 699:          cmp  #125
4aa7 9013 700:          bcc  .do_here
4aa9 a000 701: .do_look:      ldy  #0
4aab b9b24b 702: .lookup:      lda  .char_table,y
4aae f00c 703:          beq  .do_here
4ab0 c55a 704:          cmp  $5a
4ab2 f003 705:          beq  .go_cio
4ab4 c8 706:          iny
4ab5 d0f4 707:          bne  .lookup
4ab7      708:
4ab7      709:
4ab7 a55a 710: .go_cio:      lda  $5a
4ab9 4cc24b 711:          jmp  putlocal
4abc      712:
4abc      713:
4abc a000 714: .do_here:      ldy  #0

```

```

4abe a55d 715:      lda  $5d
4ac0 915e 716:      sta  ($5e),y
4ac2 a55a 717:      lda  $5a
4ac4 c99b 718:      cmp  #$9b
4ac6 f044 719:      beq  .docr
4ac8 20914b 720: .notcr:      jsr  .get_adr
4acb a55a 721:      lda  $5a
4acd 297f 722:      and  #$7f
4acf c920 723:      cmp  #32
4ad1 9007 724:      bcc  .add64
4ad3 c960 725:      cmp  #96
4ad5 9009 726:      bcc  .sub32
4ad7 4ce34a 727:      jmp  .asis
4ada 18 728: .add64:      clc
4adb 6940 729:      adc  #64
4add 4ce34a 730:      jmp  .asis
4ae0 38 731: .sub32:      sec
4ae1 e920 732:      sbc  #32
4ae3 245a 733: .asis:      bit  $5a
4ae5 1002 734:      bpl  .xxlate
4ae7 0980 735:      ora  #$80
4ae9 a000 736: .xxlate:      ldy  #0
4aeb 915e 737:      sta  ($5e),y
4aed e655 738:      inc  $55
4aef e663 739:      inc  $63
4af1 a555 740:      lda  $55
4af3 c928 741:      cmp  #40
4af5 b019 742:      bcs  .next_row
4af7 c8 743:      iny
4af8 b15e 744:      lda  ($5e),y
4afa 855d 745:      sta  $5d
4afc 0980 746:      ora  #$80
4afe aef002 747:      ldx  752
4b01 d002 748:      bne  .nocurs1
4b03 915e 749:      sta  ($5e),y
4b05 e65e 750: .nocurs1:      inc  $5e
4b07 d002 751:      bne  .nooav
4b09 e65f 752:      inc  $5e+1
4b0b 60 753: .nooav:      rts
4b0c a900 754: .docr:      lda  #0
4b0e 8563 755:      sta  $63
4b10 a552 756: .next_row:      lda  $52
4b12 8555 757:      sta  $55
4b14 e654 758:      inc  $54
4b16 a454 759:      ldy  $54
4b18 c018 760:      cpy  #24
4b1a b013 761:      bcs  .scroll
4b1c 20914b 762:      jsr  .get_adr
4b1f a000 763:      ldy  #0
4b21 b15e 764:      lda  ($5e),y
4b23 855d 765:      sta  $5d
4b25 aef002 766:      ldx  752
4b28 d004 767:      bne  .nocurs2

```



```

4b2a 0980 768:      ora  #$80
4b2c 915e 769:      sta  ($5e),y
4b2e 60 770: .nocurs2:      rts
4b2f 771:
4b2f c654 772: .scroll:      dec  $54
4b31 a558 773:      lda  $58
4b33 8568 774:      sta  $68
4b35 18 775:      clc
4b36 6928 776:      adc  #40
4b38 855e 777:      sta  $5e
4b3a a559 778:      lda  $58+1
4b3c 8569 779:      sta  $68+1
4b3e 6900 780:      adc  #0
4b40 855f 781:      sta  $5e+1
4b42 a000 782:      ldy  #0
4b44 a203 783:      ldx  #3
4b46 b15e 784: .scloop:      lda  ($5e),y
4b48 9168 785:      sta  ($68),y
4b4a c8 786:      iny
4b4b d0f9 787:      bne  .scloop
4b4d e65f 788:      inc  $5e+1
4b4f e669 789:      inc  $68+1
4b51 ca 790:      dex
4b52 d0f2 791:      bne  .scloop
4b54 b15e 792: .scloop2:      lda  ($5e),y
4b56 9168 793:      sta  ($68),y
4b58 c8 794:      iny
4b59 c098 795:      cpy  #152
4b5b 90f7 796:      bcc  .scloop2
4b5d a558 797:      lda  $58
4b5f 18 798:      clc
4b60 6998 799:      adc  #920&$ff
4b62 855e 800:      sta  $5e
4b64 a559 801:      lda  $58+1
4b66 6903 802:      adc  #920/256
4b68 855f 803:      sta  $5e+1
4b6a a000 804:      ldy  #0
4b6c 98 805:      tya
4b6d 915e 806: .clear:      sta  ($5e),y
4b6f c8 807:      iny
4b70 c028 808:      cpy  #40
4b72 90f9 809:      bcc  .clear
4b74 aef002 810:      ldx  752
4b77 d006 811:      bne  .nocurs3
4b79 a980 812:      lda  #$80
4b7b a455 813:      ldy  $55
4b7d 915e 814:      sta  ($5e),y
4b7f a900 815: .nocurs3:      lda  #0
4b81 855d 816:      sta  $5d
4b83 a55e 817:      lda  $5e
4b85 18 818:      clc
4b86 6555 819:      adc  $55
4b88 855e 820:      sta  $5e

```

```

4b8a a55f 821:      lda  $5e+1
4b8c 6900 822:      adc  #0
4b8e 855f 823:      sta  $5e+1
4b90 60   824:      rts
4b91 a554 825: .get_adr:      lda  $54
4b93 0a   826:      asl
4b94 a8   827:      tay
4b95 a558 828:      lda  $58
4b97 18   829:      clc
4b98 795f4a 830:      adc  scr_offset,y
4b9b 855e 831:      sta  $5e
4b9d a559 832:      lda  $58+1
4b9f 79604a 833:      adc  scr_offset+1,y
4ba2 855f 834:      sta  $5e+1
4ba4 a55e 835:      lda  $5e
4ba6 18   836:      clc
4ba7 6555 837:      adc  $55
4ba9 855e 838:      sta  $5e
4bab a55f 839:      lda  $5e+1
4bad 6900 840:      adc  #0
4baf 855f 841:      sta  $5e+1
4bb1 60   842:      rts
4bb2      843: .char_table:
4bb2 1b1c1d 844:      dc.b 27,28,29,30,31,125,126,127
4bba 9c9d9e 845:      dc.b 156,157,158,159,253,254,255
4bc2      846:
4bc2      847:
4bc2      848: putlocal:
4bc2 a20b 849:      ldx  #11
4bc4 8e4203 850:      stx  $0342
4bc7 a200 851:      ldx  #0
4bc9 8e4803 852:      stx  $0348
4bcc 8e4903 853:      stx  $0349
4bcf 4c56e4 854:      jmp  $e456
4bd2      855: ;
4bd2      856:
4bd2      857:
4bd2      858:
4bd2      859: ;~~~~~
4bd2      860: ;Prints a byte or a card to the screen
4bd2      861: pr_byte:
4bd2 a200 862:      ldx  #$00
4bd4      863: pr_card:
4bd4 85d4 864:      sta  fr0
4bd6 86d5 865:      stx  fr0+1
4bd8 a900 866:      lda  #$00
4bda 85f2 867:      sta  cix
4bdc 8d024c 868:      sta  .bytecard
4bdf 20aad9 869:      jsr  ifp
4be2 20e6d8 870:      jsr  fasc
4be5 a000 871:      ldy  #$00
4be7      872: .aprbloop:
4be7 ee024c 873:      inc  .bytecard

```

```

4bea b1f3 874:          lda  ($f3),y
4bec 08 875:          php
4bed 297f 876:          and  #$7f
4bef 99034c 877:         sta  .bytecard+1,y
4bf2 c8 878:          iny
4bf3 28 879:          plp
4bf4 10f1 880:         bpl  .aprbloop
4bf6      881:         mea  .bytecard,pr_ptr
4bf6 a902 ----:        lda  #low .bytecard
4bf8 8588 ----:        sta  pr_ptr
4bfa a94c ----:        lda  #high .bytecard
4bfc 8589 ----:        sta  pr_ptr + 1
4bfe 20124c 882:        jsr  echos
4c01 60 883:          rts
4c02      884:
4c02      885: .bytecard:      ds.b  10
4c0c      886:
4c0c      887:
4c0c      888:
4c0c      889: cr_lf:
4c0c 20394a 890:        jsr  printsi
4c0f 9bff 891:        dc.b  $9b,-1
4c11 60 892:          rts
4c12      893:
4c12      894:
4c12      895:
4c12      896: ; prints: prints string with address in PR_PTR, length by
4c12      897: ; ~~~~~
4c12      898:
4c12      899: echos:
4c12 a000 900:          ldy  #0
4c14 b188 901:          lda  (pr_ptr),y
4c16 f01b 902:          beq  .xpri
4c18 8d354c 903:         sta  .maxlth
4c1b c8 904:          iny
4c1c 8c344c 905:         sty  .tempy
4c1f      906:
4c1f ac344c 907: .pril:      ldy  .tempy
4c22 b188 908:          lda  (pr_ptr),y
4c24 c900 909:          cmp  #$00
4c26 f00b 910:          beq  .xpri
4c28 208f4a 911:         jsr  echo
4c2b ee344c 912: .prnext:   inc  .tempy
4c2e ce354c 913:          dec  .maxlth
4c31 d0ec 914:          bne  .pril
4c33 60 915: .xpri:      rts
4c34      916:
4c34 01 917: .tempy:      dc.b  1
4c35 01 918: .maxlth:     dc.b  1
4c36      919:
4c36      920:
4c36      921: ;
4c36      922: ; Upcase

```

```

4c36      923: ;
4c36      924: ;
4c36 c961  925: upcase:          cmp   #'a'
4c38 9009  926:          bcc   .xupcase
4c3a c97b  927:          cmp   #'z'+1
4c3c f005  928:          beq   .xupcase
4c3e b003  929:          bcs   .xupcase
4c40 38    930:          sec
4c41 e920  931:          sbc   #32
4c43 60    932: .xupcase:          rts
4c44      933:
4c44      934:
4c44 c941  935: locase:          cmp   #'A'
4c46 9009  936:          bcc   .xlocase
4c48 c95b  937:          cmp   #'Z'+1
4c4a f005  938:          beq   .xlocase
4c4c b003  939:          bcs   .xlocase
4c4e 18    940:          clc
4c4f 6920  941:          adc   #32
4c51 60    942: .xlocase:          rts
4c52      943:
4c52      944:
4c52      945:
4c52      946: ;   Make a length-byte string from an ASCII ($9b termin
4c52      947: ;   ~~~~~
4c52      948:
4c52 a000  949: tostr:          ldy   #0
4c54 b182  950: .tfe:          lda   (s_ptr),y
4c56 f007  951:          beq   .gotend
4c58 c99b  952:          cmp   #$9b
4c5a f003  953:          beq   .gotend
4c5c c8    954:          iny
4c5d d0f5  955:          bne   .tfe
4c5f 8c7a4c 956: .gotend:        sty   .lth
4c62 c000  957:          cpy   #0
4c64 f00c  958:          beq   .rtn
4c66 a000  959:          ldy   #0
4c68 b182  960: .loop:          lda   (s_ptr),y
4c6a c8    961:          iny
4c6b 9184  962:          sta   (d_ptr),y
4c6d cc7a4c 963:          cpy   .lth
4c70 90f6  964:          bcc   .loop
4c72 a000  965: .rtn:          ldy   #0
4c74 ad7a4c 966:          lda   .lth
4c77 9184  967:          sta   (d_ptr),y
4c79 60    968:          rts
4c7a      969:
4c7a 01    970: .lth:          dc.b  1
4c7b      971:
4c7b      972: ;
4c7b      973: ;   Move string pointed to by s_ptr to string pointed
4c7b      974: ;   ~~~~~
4c7b      975:

```

```

4c7b      976:
4c7b a000  977: move_str:      ldy  #0
4c7d b182  978:                lda  (s_ptr),y
4c7f a8    979:                tay
4c80 b182  980: .loop:         lda  (s_ptr),y
4c82 9184  981:                sta  (d_ptr),y
4c84 c000  982:                cpy  #0
4c86 f004  983:                beq  .done
4c88 88    984:                dey
4c89 4c804c 985:                jmp  .loop
4c8c 60    986: .done:         rts
4c8d      987:
4c8d      988:
4c8d      989:
4c8d      990: ;   Return the length of string at s_ptr in accumulat
4c8d      991: ;   ~~~~~
4c8d      992:
4c8d a000  993: length:        ldy  #0
4c8f b182  994:                lda  (s_ptr),y
4c91 60    995:                rts
4c92      996:
4c92      997:
4c92      998: ;   Make an ASCII ($9b terminated) string out of a lth-
4c92      999: ;   ~~~~~
4c92     1000:
4c92 a000 1001: toascii:       ldy  #0
4c94 b182 1002:                lda  (s_ptr),y
4c96 8db84c 1003:                sta  .lth
4c99 f019 1004:                beq  .null
4c9b c8    1005:                iny
4c9c eeb84c 1006:                inc  .lth
4c9f b182 1007: .loop:         lda  (s_ptr),y
4ca1 88    1008:                dey
4ca2 9184 1009:                sta  (d_ptr),y
4ca4 c8    1010:                iny
4ca5 c8    1011:                iny
4ca6 ccb84c 1012:                cpy  .lth
4ca9 d0f4 1013:                bne  .loop
4cab acb84c 1014: .done:         ldy  .lth
4cae 88    1015:                dey
4caf a99b 1016: .rtn:         lda  #$9b
4cb1 9184 1017:                sta  (d_ptr),y
4cb3 60    1018:                rts
4cb4 a000 1019: .null:        ldy  #0
4cb6 f0f7 1020:                beq  .rtn
4cb8      1021:
4cb8 01    1022: .lth:         dc.b  1
4cb9      1023:
4cb9      1024:
4cb9      1025:
4cb9      1026:
4cb9      1027:
4cb9     1028: win_end:      ds.b  0

```

```
4cb9    1029: ;  
4cb9    1030: buffer:          ds.b  0  
4cb9    1031: ;
```

End assembly: no errors