


```

0000 1: ;-----
0000 2: ;(Change the date in vtoc General DOS utility for real.do
0000 3: ;
0000 4: ;      Copyright 2005 Intergraded Logic Systems
0000 5: ;      Source Code is Copyright Stephen J. Car
0000 6: ;
0000 7: ;
0000 8: ;      Please do not share this source!
0000 9: ;-----
0000 10: ;      Real.dos
0000 11: ;      This program is to replace Bank_bb.COM with
0000 12: ;      bank_bb.com
0000 13: ;
0000 14: ;      Code History.
0000 15: ;
0000 16: ;Init design,      10/15/2012      sjc
0000 17: ;
0000 18: ;      ~~~~~
0000 19: ;-----
0000 20: ; Notes: o This source code MAY NOT be placed for download
0000 21: ;      o "mea" is a macro that loads the address of the
0000 22: ;      into the pointer specified by the second field.
0000 23: ;-----
0000 24: ; Assembler: MADMAC (tm) ST Cross Assembler (Atari Corp)
0000 25: ;      XASM  st and IBM VERSIONS
0000 26: ;-----
0000 27:
0000 28: com_file_name:      .macro
0000 29: ;      dc.b  13,"filename.ext",$9b
0000 30: ;      dc.b  12,"Bank_bb.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  $10
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  16
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: l_relocatable:     equ    r..no
0000      83: ;
0000      84: ;
0000      85: ;~~~~~
0000      86: ; Gotta define our Header control Byte
0000      87: ;$ff=bypass intro
0000      88: ;$15 = Bypass Dos check
0000      89: r..crl:           equ    $7d
0000      90: r..crlf:          equ    $9b
0000      91: r..space:         equ    $20
0000      92: r..bypassDOS:     equ    $15
0000      93: r..bypass:        equ    $ff
0000      94:
0000      95: l_frstscreenbyte: equ    r..crlf
0000      96: ;~~~~~
0000      97: ; The language the output file is in.
0000      98: ;
0000      99: ;
0000     100: ; 0 =   Undefined
0000     101: ; 1 =   English
0000     102: ; 2 =   German
0000     103: ;
0000     104: r..language:      equ    1
0000     105: ;~~~~~
0000     106: ;

```

```

0000      107: ;;-----
0000      108: ; pbi common Routines
0000      109: ;RAMPAGE:          equ   $D1BC
0000      110: ;CFG_PAGE:         equ   $F5
0000      111: ;MISC_PAGE:        equ   $FE
0000      112: ;DUMMYPAGE:       equ   $FF
0000      113: ;BB_Sensel:       equ   $D1C0
0000      114: ;PBIBANK:         equ   $D1FF
0000      115: ;PBISHADOW:       equ   $0248
0000      116: ;HDCODE_BANK:     equ   $04
0000      117: ;
0000      118:
0000      119:
0000      120:          .include   equates
0000      121:          .include   globals
0000      122:          .include   macros
0000      123: ;
4000      124:          .org   $4000
4000      125: ;
4000      126:
4000      127: win_start:
4000      128: header_info:
4000      129:          .include   header
42a7      130:
42a7 a03f  131:          ldy   #LBUF
42a9 b10a  132: mux_1a:          lda   (comtab),y
42ab c920  133:          cmp   #' '
42ad f00b  134:          beq   mux_1b
42af c99b  135:          cmp   #$9b
42b1 f004  136:          beq   .a
42b3 c8    137:          iny
42b4 4ca942 138:          jmp   mux_1a
42b7 4c2443 139: .a:          jmp   info          ; t
42ba      140:
42ba      141: mux_1b:
42ba a200  142:          ldx   #$00
42bc c8    143:          iny
42bd      144:
42bd b10a  145: .a:          lda   (comtab),y
42bf 209243 146:          jsr   upcase
42c2 c99b  147:          cmp   #$9b
42c4 f00d  148:          beq   .noparm
42c6 c944  149:          cmp   #'D'
42c8 f00c  150:          beq   .got_d
42ca      151:
42ca      152:
42ca      153:
42ca e8    154:          inx
42cb 8ed242 155:          stx   .holdx
42ce c8    156:          iny
42cf 4cbd42 157:          jmp   .a
42d2      158:
42d2 00    159: .holdx:          dc.b   0

```

```

42d3      160:
42d3 4c2443 161: .noparm:      jmp  info      ;te
42d6      162:
42d6 c8    163: .got_d:      iny
42d7 b10a   164:      lda  (comtab),y
42d9 c930   165:      cmp  #'1'-1
42db 90f6   166:      bcc  .noparm
42dd c93a   167:      cmp  #'9'+1
42df b0f2   168:      bcs  .noparm
42e1 8d9d43 169:      sta  my_drive
42e4 e930   170:      sbc  #48
42e6 8d9e43 171:      STA  Black_box_drv
42e9 a200   172:      ldx  #$00
42eb      173: ;      inx
42eb c8    174:      iny
42ec c8    175:      iny
42ed c8    176:      iny
42ee b10a   177:      lda  (comtab),y
42f0 9df743 178:      sta  partition_ascii,x
42f3      179: ;      jsr  echo
42f3 c99b   180:      cmp  #$9b
42f5 f01b   181:      beq  .done
42f7 c92f   182:      cmp  #'0'-1
42f9 90d8   183:      bcc  .noparm
42fb c93a   184:      cmp  #'9'+1
42fd b0d4   185:      bcs  .noparm
42ff c8    186:      iny
4300 e8    187:      inx
4301 b10a   188:      lda  (comtab),y
4303 9df743 189:      sta  partition_ascii,x
4306      190: ;      jsr  echo
4306 c99b   191:      cmp  #$9b
4308 f008   192:      beq  .done
430a c92f   193:      cmp  #'0'-1
430c 90c5   194:      bcc  .noparm
430e c93a   195:      cmp  #'9'+1
4310 b0c1   196:      bcs  .noparm
4312      197: .done:
4312 20c443 198:      jsr  get_number_cli
4315 8dfd43 199:      sta  partition_number
4318 c960   200:      cmp  #96
431a b008   201:      bcs  info
431c c900   202:      cmp  #$00
431e f004   203:      beq  info
4320      204: ;      lda  #$9b
4320      205: ;      jsr  echo
4320      206: ;      lda  partition_number
4320      207: ;      jsr  pr_byte
4320      208: ;
4320      209: ;      lda  #$9b
4320      210: ;      jsr  echo
4320      211: ;      lda  Black_box_drv
4320      212: ;      jsr  pr_byte

```

```

4320      213:
4320 4cf43 214:          jmp    start
4323      215:
4323      216:
4323 00    217: .myhold:          dc.b    0
4324      218: ;
4324      219:
4324      220: info:
4324 201045 221:          JSR    printsi
4327 9b9b  222:          dc.b    $9b,$9b
4329 c2ece1 223:          dc.b    <+128>,"Black Box all Types
434a 9b    224:          dc.b    $9b
434b 557361 225:          dc.b    "Usage:",$9B
4352 42616e 226:          dc.b    "Bank_bb Dn: <Partition>",$
436a 202044 227:          dc.b    " Dn: n=1 to 9",$9b
4379 506172 228:          dc.b    "Partition = 1 to 96",$9b
438d 9b9b9b 229:          dc.b    $9b,$9b,$9b,-1
4391 60    230:          rts
4392      231:
4392      232:
4392      233:
4392      234: ;-----
4392      235: ;CHANGES THE CHARTER TO UPGASE
4392      236: ;-----
4392      237: upcase:
4392 c961    238:          cmp     #$61
4394 9006    239:          bcc     .a
4396 c97a    240:          cmp     #$7a
4398 b002    241:          bcs     .a
439a e91f    242:          sbc     #$1f
439c 60     243: .a:          rts
439d      244:
439d      245:
439d      246:
439d 00     247: my_drive:          dc.b    0
439e      248: ;
439e      249:
439e 00     250: Black_box_drv:          dc.b    0
439f      251:
439f      252: partition_info:          ds.b    9
43a8      253:          ds.b    1          ; b
43a9      254: par_density:          ds.b    1          ; B
43aa      255:          ds.b    4          ; a
43ae      256: par_sect_size:          ds.w    1          ; n
43b0      257:          ds.b    1
43b1      258:          ds.b    1
43b2      259: ;
43b2      260: my_drive_lo:
43b2 00     261:          dc.b    BBHD1
43b3 12     262:          dc.b    BBHD2
43b4 24     263:          dc.b    BBHD3
43b5 36     264:          dc.b    BBHD4
43b6 48     265:          dc.b    BBHD5

```

```

43b7 5a 266:          dc.b  BBHD6
43b8 6c 267:          dc.b  BBHD7
43b9 7e 268:          dc.b  BBHD8
43ba 90 269:          dc.b  BBHD9
43bb    270:
43bb    271: my_drive_high:
43bb d6 272:          dc.b  BBHD1 / 256
43bc d6 273:          dc.b  BBHD2 / 256
43bd d6 274:          dc.b  BBHD3 / 256
43be d6 275:          dc.b  BBHD4 / 256
43bf d6 276:          dc.b  BBHD5 / 256
43c0 d6 277:          dc.b  BBHD6 / 256
43c1 d6 278:          dc.b  BBHD7 / 256
43c2 d6 279:          dc.b  BBHD8 / 256
43c3 d6 280:          dc.b  BBHD9 / 256
43c4    281:
43c4    282: ;
43c4    283: ;
43c4    284: ;   gets a CARD value - returned in A/X
43c4    285: ;   ~~~~~
43c4    286:
43c4    287: get_number_cli:
43c4    288:      mea  partition_ascii,inbuff
43c4 a9f7 ----:      lda  #low partition_ascii
43c6 85f3 ----:      sta  inbuff
43c8 a943 ----:      lda  #high partition_ascii
43ca 85f4 ----:      sta  inbuff + 1
43cc a900 289:      lda  #0          ;l
43ce 85f2 290:      sta  cix
43d0 2000d8 291:      jsr  afp          ;as
43d3 b00a 292:      bcs  .default
43d5 20d2d9 293:      jsr  fpi          ;fl
43d8 b005 294:      bcs  .default
43da a5d4 295:      lda  fr0
43dc a6d5 296:      ldx  fr0+1
43de 60 297:      rts
43df a900 298: .default:      lda  #0
43e1 aa 299:      tax
43e2 60 300:      rts
43e3    301:
43e3    302: .work:      ds.b  20
43f7    303:
43f7    304:
43f7 009b9b 305: partition_ascii:      dc.b  0,$9b,$9b,$9b,$9b,$
43fd 00 306: partition_number:      dc.b  0
43fe    307: ;
43fe    308: start:
43fe 207945 309:      jsr  black_box
4401 adb145 310:      lda  pbi_type
4404 c902 311:      cmp  #$02
4406 f003 312:      beq  .a
4408 4c2443 313:      jmp  info
440b    314: .a:

```

```

440b      315:
440b ac9e43 316:      ldy  Black_box_drv
440e b9b243 317:      lda  my_drive_lo,y
4411 8d5244 318:      sta  bb_set_drv+1
4414 b9bb43 319:      lda  my_drive_high,y
4417 8d5344 320:      sta  bb_set_drv+2
441a      321:
441a      322: ;
441a adfd43 323:      LDA  partition_number
441d      324: ;
441d      325: SET_PART:
441d 18      326:      CLC
441e 690f 327:      ADC  #$0F      ;th
4420 a2fe 328:      LDX  #MISC_PAGE ;se
4422 8ebcd1 329:      STX  RAMPAGE
4425 8d53d6 330:      STA  $D653      ;we
4428      331: ;
4428 a904 332:      LDA  #HDCODE_BANK ;ba
442a 8d4802 333:      STA  PBISHADOW ;s
442d 8dffd1 334:      STA  PBIBANK ;pu
4430 202cd8 335:      JSR  $D82C      ;**
4433 a900 336:      LDA  #0
4435 8d4802 337:      STA  PBISHADOW
4438 8dffd1 338:      STA  PBIBANK
443b b022 339:      BCS  SET_ERROR
443d ad56d6 340:      LDA  $D656      ;if
4440 301d 341:      BMI  SET_ERROR ;pa
4442 a000 342:      LDY  #0      ;pr
4444 a2fe 343: SET_1:      LDX  #MISC_PAGE ;ge
4446 8ebcd1 344:      STX  RAMPAGE ;pa
4449 b956d6 345:      LDA  $D656,Y ;mi
444c a2f5 346:      LDX  #CFG_PAGE ;no
444e 8ebcd1 347:      STX  RAMPAGE ;co
4451      348: bb_set_drv:
4451 99ffff 349:      STA  $ffff,Y ;$d
4454 999f43 350:      sta  partition_info,y
4457 c8      351:      INY
4458 c012 352:      CPY  #18      ;c
445a 90e8 353:      BCC  SET_1
445c 18      354:      CLC      ;We
445d 9001 355:      BCC  SET_2      ;sk
445f 38      356: SET_ERROR:      SEC      ;se
4460 a9ff 357: SET_2:      LDA  #DUMMYPAGE ;re
4462 8dbcd1 358:      STA  RAMPAGE
4465      359:
4465 201045 360:      jsr  printsi
4468 9b9b 361:      dc.b $9b,$9b
446a 506172 362:      dc.b "Partition",-1
4475      363:
4475 a922 364:      lda  #""
4477 20f744 365:      jsr  echo
447a ad9f43 366:      lda  partition_info
447d c900 367:      cmp  #$00

```



```

447f f003 368:      beq  .n2
4481 20f744 369:      jsr  echo
4484      370: .n2:
4484 ada043 371:      lda  partition_info+1
4487 c900 372:      cmp  #$00
4489 f003 373:      beq  .n3
448b 20f744 374:      jsr  echo
448e      375: .n3:
448e ada143 376:      lda  partition_info+2
4491 c900 377:      cmp  #$00
4493 f003 378:      beq  .n4
4495 20f744 379:      jsr  echo
4498      380: .n4:
4498 ada243 381:      lda  partition_info+3
449b c900 382:      cmp  #$00
449d f003 383:      beq  .n5
449f 20f744 384:      jsr  echo
44a2      385:
44a2      386: .n5:
44a2 ada343 387:      lda  partition_info+4
44a5 c900 388:      cmp  #$00
44a7 f003 389:      beq  .n6
44a9 20f744 390:      jsr  echo
44ac      391:
44ac      392: .n6:
44ac ada443 393:      lda  partition_info+5
44af c900 394:      cmp  #$00
44b1 f003 395:      beq  .n7
44b3 20f744 396:      jsr  echo
44b6      397:
44b6      398: .n7:
44b6 ada543 399:      lda  partition_info+6
44b9 c900 400:      cmp  #$00
44bb f003 401:      beq  .n8
44bd 20f744 402:      jsr  echo
44c0      403:
44c0      404: .n8:
44c0 ada643 405:      lda  partition_info+7
44c3 c900 406:      cmp  #$00
44c5 f003 407:      beq  .n9
44c7 20f744 408:      jsr  echo
44ca      409:
44ca      410: .n9:
44ca ada743 411:      lda  partition_info+8
44cd c900 412:      cmp  #$00
44cf f003 413:      beq  .na
44d1 20f744 414:      jsr  echo
44d4      415: .na:
44d4 a922 416:      lda  #""
44d6 20f744 417:      jsr  echo
44d9      418:
44d9 201045 419:      jsr  printsi
44dc 204d61 420:      dc.b  " Mapped to D",-1

```

```

44e9 ad9d43 421:          lda  my_drive
44ec 20f744 422:          jsr  echo
44ef 201045 423:          jsr  printsi
44f2 3a9b9b 424:          dc.b  ":",$9b,$9b,-1
44f6      425:
44f6      426:
44f6 60    427:          RTS          ; a
44f7      428: ;
44f7      429: ;
44f7      430: ;=====
44f7      431: ;   print/input routines
44f7      432: ;=====
44f7      433: ;
44f7      434: ;   Print character
44f7      435: ;   -----
44f7      436: ; in:
44f7      437: ;   A   = character to print
44f7      438: ; out:
44f7      439: ;   all registers preserved
44f7      440: ;
44f7 204245 441: ECHO:      JSR  SAVER
44fa 200045 442:          JSR  ZOUT
44fd 4c7245 443:          JMP  RESALL
4500      444: ;
4500 8d0c45 445: ZOUT:      STA  ZTEMP+1
4503 ad4703 446:          LDA  $347
4506 48    447:          PHA
4507 ad4603 448:          LDA  $0346
450a 48    449:          PHA
450b a900   450: ZTEMP:    LDA  #0
450d a200   451:          LDX  #0          ;1   ; force NO
450f 60    452:          RTS
4510      453: ;
4510      454: ;
4510      455: ;   Print text
4510      456: ;   -----
4510      457: ; out:
4510      458: ;   all registers preserved
4510      459: ; notes:
4510      460: ;   This print routine will print all characters
4510      461: ;   following the JSR PRINT until a delimiter of
4510      462: ;   -1 ($FF) is reached. PRINT will return to the
4510      463: ;   point one byte beyond the delimiter.
4510      464: ;
4510 204245 465: printsi: JSR  SAVER
4513 ba    466:          TSX
4514 bd0501 467:          LDA  $0105,X
4517 8d2345 468:          STA  ZOCH+1
451a bd0601 469:          LDA  $0106,X
451d 8d2445 470:          STA  ZOCH+2
4520      471: ;
4520 a001   472:          LDY  #1
4522 b9ffff 473: ZOCH:      LDA  $FFFF,Y

```

```

4525 c9ff 474:      CMP  #$FF
4527 f006 475:      BEQ  ZECHO
4529 20f744 476:     JSR  ECHO
452c c8 477:      INY
452d d0f3 478:      BNE  ZOCH
452f 98 479: ZECHO:   TYA
4530 18 480:      CLC
4531 7d0501 481:     ADC  $0105,X
4534 9d0501 482:     STA  $0105,X
4537 bd0601 483:     LDA  $0106,X
453a 6900 484:     ADC  #0
453c 9d0601 485:     STA  $0106,X
453f 4c7245 486:     JMP  RESALL
4542      487: ;
4542      488: ;
4542      489: ;   SAVE & RESTORE registers
4542      490: ;   -----
4542      491: ;
4542 08 492: SAVER:   PHP
4543 48 493:      PHA
4544 48 494:      PHA
4545 48 495:      PHA
4546 08 496:      PHP
4547 48 497:      PHA
4548 8a 498:      TXA
4549 48 499:      PHA
454a ba 500:      TSX
454b bd0901 501:     LDA  $0109,X
454e 9d0501 502:     STA  $0105,X
4551 bd0701 503:     LDA  $0107,X
4554 9d0901 504:     STA  $0109,X
4557 bd0101 505:     LDA  $0101,X
455a 9d0701 506:     STA  $0107,X
455d bd0801 507:     LDA  $0108,X
4560 9d0401 508:     STA  $0104,X
4563 bd0601 509:     LDA  $0106,X
4566 9d0801 510:     STA  $0108,X
4569 98 511:      TYA
456a 9d0601 512:     STA  $0106,X
456d 68 513:      PLA
456e aa 514:      TAX
456f 68 515:      PLA
4570 28 516:      PLP
4571 60 517:      RTS
4572      518: ;
4572 68 519: RESALL:   PLA
4573 a8 520:      TAY
4574 68 521:      PLA
4575 aa 522:      TAX
4576 68 523:      PLA
4577 28 524:      PLP
4578 60 525:      RTS
4579      526: ;

```

```

4579      527: ;
4579      528: ; Checking on the black box if it is present!
4579      529: ;
4579      530: BLACK_BOX:
4579 78    531:          SEI
457a a902  532:          LDa  #2
457c 8dc0d1 533:          STa  BB_Sensel
457f a010  534:          ldy  #$10
4581      535: .a
4581 a202  536:          ldx  #2
4583 b900d8 537:          LDa  bb_rom,Y
4586 ddad45 538:          cmp  .e,X
4589 f007  539:          beq  .b
458b c8    540:          INY
458c d0f3  541:          bne  .a
458e 4ca545 542:          jmp  .d
4591 60    543:          RTS
4592      544: ;
4592      545: .b
4592 c8    546:          INY
4593 ca    547:          DEX
4594 300a  548:          bmi  .c
4596 b900d8 549:          LDa  bb_rom,Y
4599 ddad45 550:          cmp  .e,X
459c d0e3  551:          bne  .a
459e f0f2  552:          beq  .b
45a0      553: .c
45a0 a902  554:          lda  #$02
45a2 8db145 555:          sta  pbi_type
45a5      556:
45a5      557: ;
45a5      558: .d
45a5 d8    559:          CLD
45a6 a900  560:          LDa  #0
45a8 8dc0d1 561:          STa  BB_Sensel
45ab 58    562:          CLI
45ac 60    563:          RTS
45ad      564: ;
45ad d00829 565: .e          dc.b  $D0,$08,$29,$00
45b1      566: ;
45b1      567: ;-----
45b1      568: ; 1= mio
45b1      569: ; 2= Black Box
45b1      570: ; 3= kpi
45b1      571: ; 4= IDEa
45b1      572: ;
45b1 00    573: pbi_type:      dc.b  0
45b2      574: ;=====
45b2      575: ;~~~~~
45b2      576: ;Prints a byte or a card to the screen
45b2      577: ;
45b2      578: ;
45b2      579: pr_byte:

```

```
45b2 a200 580:      ldx  #$00
45b4      581: pr_card:
45b4 85d4 582:      sta  fr0
45b6 86d5 583:      stx  fr0+1
45b8 a900 584:      lda  #$00
45ba 85f2 585:      sta  cix
45bc 20aad9 586:     jsr  ifp
45bf 20e6d8 587:     jsr  fasc
45c2 a000 588:      ldy  #$00
45c4      589: .aprbloop:
45c4 b1f3 590:      lda  (inbuff),y
45c6 08 591:      php
45c7 297f 592:      and  #$7f
45c9 c8 593:      iny
45ca 20f744 594:     jsr  echo
45cd 28 595:      plp
45ce 10f4 596:      bpl  .aprbloop
45d0 60 597:      rts
45d1      598:
45d1      599:
45d1      600: ;
45d1      601: win_end:      ds.b  0
45d1      602: ;
```

End assembly: no errors