

Electronic Arts Fat Tracks File Buster

By Fungus/Nostalgia. The binary can be downloaded from [this page on CSDb](#).

```
;-----  
;EA Fat Tracks File Buster v1.0  
;  
;Code by Fungus of Nostalgia 11-2010  
;  
;All code (C) 2010 Nostalgia  
;-----  
  
    * = $c000  
  
talk      = $ffb4  
tksa      = $ff96  
untlk     = $ffab  
acptr     = $ffa5  
ciout     = $ffa8  
open      = $ffc0  
close     = $ffc3  
listen    = $ffb1  
unlsn     = $ffae  
second    = $ff93  
chkin     = $ffc6  
chkout    = $ffc9  
clrchn    = $ffcc  
chrin     = $ffcf  
chrout    = $ffd2  
getin     = $ffe4  
clall     = $ffe7  
setlfs    = $ffba  
setnam    = $ffbd  
load      = $ffd5  
save      = $ffd8  
  
check     = $02  
crypt1    = $2a  
crypt2    = $2b  
crypt3    = $2c  
loadaddr  = $2d  
temp1     = $fb  
temp2     = $fc  
temp3     = $fd  
temp4     = $fe  
temp5     = $ff  
temp6     = $03  
temp7     = $04
```

```
status      = $0609
statusmessage = $0612
blocksinfile = $0541
blocks      = $0552
loadingblock = $0569
lblock      = $057a
savingblock = $0591
sblock      = $05a2
entrypoint  = $05b9
entry       = $05cb
versionmessage = $04f1
```

```
    lda #$00
    sta $0289
    sta $0286
    lda #$c1
    sta $0318
    lda #$80
    sta $0291
    jsr setupscreen
```

main

```
    lda #$00
    sta $9d
    sta block
    lda #$ef
```

waitspc

```
    cmp $dc01
    bne waitspc
```

```
    jsr loadea
    lda $1003
    sta loadhigh
    lda $1004
    sta loadend
    lda $1005
    sta checksum
    lda $1004
    sec
    sbc $1003
    sta sblocks
    jsr printblocks
    lda sblocks
    jsr hex2dec
    sty $0552
    stx $0553
    sta $0554
```

```
    lda $1001
    cmp #$08
```

```
    beq isver1
    cmp #$12
    beq checkver
    jmp versionerror
```

isver1

```
    lda $126d
    sta start
    lda $126e
    sta start+$01
    lda #$01
    sta version
    jmp begin
```

checkver

```
    lda $12ab
    cmp #$4c
    beq isver2
    lda $12b5
    cmp #$4c
    beq isver3
    lda $12c0
    cmp #$4c
    beq isver4
    lda $12ee
    cmp #$4c
    beq isver5
    jmp versionerror
```

isver2

```
    jsr fixstart
    lda $12ac
    sec
    sbc temp1
    sta start
    lda $12ad
    sbc temp2
    sta start+$01
    jsr loadcopy
    lda #$02
    sta version
    jmp begin
```

isver3

```
    jsr fixstart
    lda $12b6
    sec
    sbc temp1
    sta start
    lda $12b7
```

```
sbc temp2
sta start+$01
jsr loadcopy
lda #$03
sta version
jmp begin
```

isver4

```
jsr fixstart
lda $12c1
sec
sbc temp1
sta start
lda $12c2
sbc temp2
sta start+$01
jsr loadcopy
lda #$04
sta version
jmp begin
```

isver5

```
jsr fixstart
lda $12ef
sec
sbc temp1
sta start
lda $12f0
sbc temp2
sta start+$01
jsr loadcopy
lda #$05
sta version
```

begin

```
lda version
ora #$30
sta versiondetected+$0c

jsr printversion
jsr printentry

lda start+$01
jsr hex2pet
sty $05cb
sta $05cc
lda start
jsr hex2pet
sty $05cd
```

```
sta $05ce

ldx #$30
stx blockread+$07
stx blockread+$0a
stx blockread+$0b
inx
stx blockread+$08
```

```
jsr fillmem
jsr loadfile
```

```
lda version
cmp #$01
beq type1c
cmp #$02
beq type2c
cmp #$03
beq type3c
cmp #$04
beq type4c
cmp #$05
beq type5c
jmp versionerror
```

```
type1c
sei
jsr dodecrypt1
jsr dochecksum
cli
beq dosave
jmp checkerror
```

```
type2c
sei
jsr getcrypt2
jsr dodecrypt3
jsr dochecksum
cli
beq dosave
jmp checkerror
```

```
type3c
sei
jsr getcrypt3
jsr dodecrypt4
jsr dochecksum
cli
beq dosave
jmp checkerror
```

type4c

```
sei
jsr getcrypt4
jsr dodecrypt4
jsr dochecksum
cli
beq dosave
jmp checkerror
```

type5c

```
sei
jsr getcrypt1
jsr dodecrypt2
jsr dochecksum
cli
beq dosave
jmp checkerror
```

dosave

```
jsr savefile
jsr printdone
jmp main
```

loadfile

```
jsr printloading
jsr clall
jsr setstart
jsr basicout
lda #$03
ldx #<initialize
ldy #>initialize
jsr setnam
lda #$0f
tay
ldx #$08
jsr setlfs
jsr open
lda #$01
ldx #<channel
ldy #>channel
jsr setnam
lda #$02
tay
ldx #$08
jsr setlfs
jsr open
```

loadloop

```
inc block
```

```
    lda block
    jsr hex2dec
    sty $057a
    stx $057b
    sta $057c
    jsr sendbr
    jsr sendbp
    jsr loadblock
    inc loadaddr+$01
    lda loadaddr+$01
    cmp loadend
    bne loadloop
eof
    lda #$02
    jsr close
    lda #$0f
    jsr close
    jsr clall
    rts

sendbr
    lda #$08
    jsr listen
    lda #$6f
    jsr second
    ldy #$00
brloop lda blockread,y
    beq brexit
    jsr ciout
    iny
    bne brloop
brexit jsr unlsn
    rts

sendbp
    lda #$08
    jsr listen
    lda #$6f
    jsr second
    ldy #$00
bploop lda bufferpointer,y
    beq bpexit
    jsr ciout
    iny
    bne bploop
bpexit jsr unlsn
    rts

loadblock
```

```
    lda #$08
    jsr talk
    lda #$62
    jsr tksa
    ldy #$00
lbloop jsr acptr
    pha
    lda $90
    and #$bf
    beq byteok
    pla
    pla
    pla
    jsr eof
    jmp loaderror
byteok
    pla
    inc $d020
    sta (loadaddr),y
    dec $d020
    iny
    bne lbloop
    jsr untlk
    inc blockread+$0b
    lda blockread+$0b
    cmp #$3a
    bne lbexit
    lda #$30
    sta blockread+$0b
    inc blockread+$0a
    lda blockread+$0a
    cmp #$32
    bne lbexit
    lda #$30
    sta blockread+$0a
    inc blockread+$08
    lda blockread+$08
    cmp #$3a
    bne lbexit
    lda #$30
    sta blockread+$08
    inc blockread+$07
lbexit
    rts

fixstart

    lda #$00
    sta temp1
    sta temp2
```



```
    lda $1004
    clc
    adc $1005
    sta temp1
    bcc ioll1
    inc temp2
ioll1
    rts

loadcopy
    ldx #$00
copyload
    lda $1000,x
    sta $e000,x
    lda $1100,x
    sta $e100,x
    lda $1200,x
    sta $e200,x
    lda $1300,x
    sta $e300,x
    lda $1400,x
    sta $e400,x
    lda $1500,x
    sta $e500,x
    lda $1600,x
    sta $e600,x
    inx
    bne copyload
    rts
```

dochecksum

```
    jsr setstart
    jsr basicout
    ldy #$00
    sty check
chkloop lda (loadaddr),y
    clc
    adc check
    sta check
    inc loadaddr
    bne chkloop
    inc loadaddr+$01
    lda loadaddr+$01
    cmp loadend
    bne chkloop
    jsr basicin
    lda check
    sec
    sbc checksum
    rts
```

dodecrypt1

```
lda #$00
sta loadaddr
tay
lda #$20
sta loadaddr+$01
lda #$01
sta crypt1
sta crypt3
lda #$03
sta crypt2
```

cryptloop1

```
lda (loadaddr),y
eor crypt3
sta (loadaddr),y
inc loadaddr
lda (loadaddr),y
eor crypt3
sta (loadaddr),y
inc loadaddr
bne coll1
inc loadaddr+$01
```

coll1

```
lda loadaddr+$01
eor #$7f
sta crypt3
```

```
lda (loadaddr),y
eor crypt3
sta (loadaddr),y
inc loadaddr
lda (loadaddr),y
eor crypt3
sta (loadaddr),y
inc loadaddr
bne coll2
inc loadaddr+$01
```

coll2

```
lda loadaddr+$01
eor #$7f
sta crypt3
```

```
lda (loadaddr),y
eor crypt3
sta (loadaddr),y
inc loadaddr
lda (loadaddr),y
eor crypt3
```

```
    sta (loadaddr),y
    inc loadaddr
    bne coll3
    inc loadaddr+$01
coll3
    lda loadaddr+$01
    eor #$7f
    sta crypt3
    lda (loadaddr),y
    eor crypt3
    sta (loadaddr),y
    inc loadaddr
    lda (loadaddr),y
    eor crypt3
    sta (loadaddr),y
    inc loadaddr
    bne coll4
    inc loadaddr+$01
coll4
    lda loadaddr+$01
    eor #$7f
    sta crypt3
    beq checkend1

    inc crypt1
    lda crypt1
    cmp crypt2
    beq coll5
    sta crypt3
    jmp cryptloop1
coll5
    lda #$01
    sta crypt1
    inc crypt2
    lda crypt2
    sta crypt3
    jmp cryptloop1

checkend1
    rts

dodecrypt2
    lda #$00
    sta loadaddr
    tay
    lda #$08
    sta loadaddr+$01
    lda cryptkey
    sta crypt1
    sta crypt3
    lda cryptkey+$01
```

```
sta crypt2
```

```
cryptloop2
```

```
lda (loadaddr),y  
eor crypt3  
sta (loadaddr),y  
inc loadaddr  
lda (loadaddr),y  
eor crypt3  
sta (loadaddr),y  
inc loadaddr  
bne doll1  
inc loadaddr+$01
```

```
doll1
```

```
lda loadaddr+$01  
eor #$7f  
sta crypt3
```

```
lda (loadaddr),y  
eor crypt3  
sta (loadaddr),y  
inc loadaddr  
lda (loadaddr),y  
eor crypt3  
sta (loadaddr),y  
inc loadaddr  
bne doll2  
inc loadaddr+$01
```

```
doll2
```

```
lda loadaddr+$01  
eor #$7f  
sta crypt3
```

```
lda (loadaddr),y  
eor crypt3  
sta (loadaddr),y  
inc loadaddr  
lda (loadaddr),y  
eor crypt3  
sta (loadaddr),y  
inc loadaddr  
bne doll3  
inc loadaddr+$01
```

```
doll3
```

```
lda loadaddr+$01  
eor #$7f  
sta crypt3  
lda (loadaddr),y  
eor crypt3
```

```
    sta (loadaddr),y
    inc loadaddr
    lda (loadaddr),y
    eor crypt3
    sta (loadaddr),y
    inc loadaddr
    bne doll4
    inc loadaddr+$01
doll4
    lda loadaddr+$01
    eor #$7f
    sta crypt3
    beq checkend2

    dec crypt1
    lda crypt1
    cmp crypt2
    beq doll5
    sta crypt3
    jmp cryptloop2
doll5
    lda #$f1
    sta crypt1
    inc crypt2
    lda crypt2
    sta crypt3
    jmp cryptloop2

checkend2

    rts

dodecrypt3
    lda #$00
    sta loadaddr
    tay
    lda #$20
    sta loadaddr+$01
    lda cryptkey
    sta crypt1
    sta crypt3
    lda cryptkey+$01
    sta crypt2

cryptloop3

    lda (loadaddr),y
    eor crypt3
    sta (loadaddr),y
    inc loadaddr
    lda (loadaddr),y
```

```
    eor crypt3
    sta (loadaddr),y
    inc loadaddr
    bne eoll1
    inc loadaddr+$01
eoll1
    lda loadaddr+$01
    eor #$7f
    sta crypt3

    lda (loadaddr),y
    eor crypt3
    sta (loadaddr),y
    inc loadaddr
    lda (loadaddr),y
    eor crypt3
    sta (loadaddr),y
    inc loadaddr
    bne eoll2
    inc loadaddr+$01
eoll2
    lda loadaddr+$01
    eor #$7f
    sta crypt3

    lda (loadaddr),y
    eor crypt3
    sta (loadaddr),y
    inc loadaddr
    lda (loadaddr),y
    eor crypt3
    sta (loadaddr),y
    inc loadaddr
    bne eoll3
    inc loadaddr+$01
eoll3
    lda loadaddr+$01
    eor #$7f
    sta crypt3
    lda (loadaddr),y
    eor crypt3
    sta (loadaddr),y
    inc loadaddr
    lda (loadaddr),y
    eor crypt3
    sta (loadaddr),y
    inc loadaddr
    bne eoll4
    inc loadaddr+$01
eoll4
```

```
    lda loadaddr+$01
    eor #$7f
    sta crypt3
    beq checkend3

    dec crypt1
    lda crypt1
    cmp crypt2
    beq eoll5
    sta crypt3
    jmp cryptloop3
eoll5
    lda #$f1
    sta crypt1
    inc crypt2
    lda crypt2
    sta crypt3
    jmp cryptloop3

checkend3

    rts

dodecrypt4
    lda #$00
    sta loadaddr
    tay
    lda #$20
    sta loadaddr+$01
    lda cryptkey
    sta crypt1
    sta crypt3
    lda cryptkey+$01
    sta crypt2

cryptloop4

    lda (loadaddr),y
    eor crypt3
    sta (loadaddr),y
    inc loadaddr
    lda (loadaddr),y
    eor crypt3
    sta (loadaddr),y
    inc loadaddr
    bne foll1
    inc loadaddr+$01
foll1
    lda loadaddr+$01
    eor #$7f
    sta crypt3
```

```
    lda (loadaddr),y
    eor crypt3
    sta (loadaddr),y
    inc loadaddr
    lda (loadaddr),y
    eor crypt3
    sta (loadaddr),y
    inc loadaddr
    bne foll2
    inc loadaddr+$01
foll2
    lda loadaddr+$01
    eor #$7f
    sta crypt3

    lda (loadaddr),y
    eor crypt3
    sta (loadaddr),y
    inc loadaddr
    lda (loadaddr),y
    eor crypt3
    sta (loadaddr),y
    inc loadaddr
    bne foll3
    inc loadaddr+$01
foll3
    lda loadaddr+$01
    eor #$7f
    sta crypt3
    lda (loadaddr),y
    eor crypt3
    sta (loadaddr),y
    inc loadaddr
    lda (loadaddr),y
    eor crypt3
    sta (loadaddr),y
    inc loadaddr
    bne foll4
    inc loadaddr+$01
foll4
    lda loadaddr+$01
    eor #$7f
    sta crypt3
    beq checkend4

    dec crypt1
    lda crypt1
    cmp crypt2
    beq foll5
    sta crypt3
```



```
    jmp cryptloop4
foll5
    lda #$ee
    sta crypt1
    inc crypt2
    lda crypt2
    sta crypt3
    jmp cryptloop4

checkend4

    rts

getcrypt1

    ldy #$07
    ldx #$05

copyblock1

    lda blockread,y
    sta $e2f6,y
    iny
    dex
    bne copyblock1

    lda #$ce
    sta $e4d3
    lda #$f1
    sta $e4e0
    jsr docrypt1
    dec cryptkey
    lda cryptkey
    cmp cryptkey+$01
    bne gcexit1
    lda #$f1
    sta cryptkey
    inc cryptkey+$01
gcexit1
    rts

docrypt1

    sei
    jsr romsout
    lda #$08
    sta temp1
    lda #$e0
    sta temp2
    lda #$e6
    sta temp3
```

```
    lda #$02
    sta temp4
    jsr getkey1
    clc
    adc loadend
    sta cryptkey
    lda #$09
    sta temp1
    lda #$e3
    sta temp2
    lda #$27
    sta temp3
    lda #$02
    sta temp4
    jsr getkey1
    sta cryptkey+$01
    jsr romsin
    cli
    rts
getkey1
    ldy #$00
    tya
gclloop1
    asl a
    adc (temp1),y
    cpy temp3
    bne goll1
    dec temp4
    bpl goll1
    lda cryptkey+$01
    rts

goll1
    sta cryptkey+$01
    iny
    bne gclloop1
    inc temp2
    jmp gclloop1

getcrypt2

    ldy #$07
    ldx #$05

copyblock2

    lda blockread,y
    sta $e2b6,y
    iny
```

```
    dex
    bne copyblock2

    lda #$ce
    sta $e492
    lda #$f1
    sta $e49f
    jsr docrypt2
    dec cryptkey
    lda cryptkey
    cmp cryptkey+$01
    bne gcexit2
    lda #$f1
    sta cryptkey
    inc cryptkey+$01
gcexit2
    rts

docrypt2

    sei
    jsr romsout
    lda #$08
    sta temp1
    lda #$e0
    sta temp2
    lda #$a3
    sta temp3
    lda #$02
    sta temp4
    jsr getkey2
    clc
    adc loadend
    sta cryptkey
    lda #$c6
    sta temp1
    lda #$e2
    sta temp2
    lda #$29
    sta temp3
    lda #$02
    sta temp4
    jsr getkey2
    sta cryptkey+$01
    jsr romsin
    cli
    rts
getkey2
    ldy #$00
    tya
gclloop2
```

```
    asl a
    adc (temp1),y
    cpy temp3
    bne goll2
    dec temp4
    bpl goll2
    lda cryptkey+$01
    rts
```

```
goll2
    sta cryptkey+$01
    iny
    bne gcloop2
    inc temp2
    jmp gcloop2
```

```
getcrypt3
```

```
    ldy #$07
    ldx #$05
```

```
copyblock3
```

```
    lda blockread,y
    sta $e2c0,y
    iny
    dex
    bne copyblock3
```

```
    lda #$ce
    sta $e44d
    lda #$ee
    sta $e45a
    jsr docrypt3
    dec cryptkey
    lda cryptkey
    cmp cryptkey+$01
    bne gcexit3
    lda #$ee
    sta cryptkey
    inc cryptkey+$01
```

```
gcexit3
```

```
    rts
```

```
docrypt3
```

```
    sei
    jsr romsout
    lda #$08
    sta temp1
```

```
    lda #$e0
    sta temp2
    lda #$ad
    sta temp3
    lda #$02
    sta temp4
    jsr getkey3
    clc
    adc loadend
    sta cryptkey
    lda #$d0
    sta temp1
    lda #$e2
    sta temp2
    lda #$da
    sta temp3
    lda #$01
    sta temp4
    jsr getkey3
    sta cryptkey+$01
    jsr romsin
    cli
    rts
getkey3
    ldy #$00
    tya
gloop3
    asl a
    adc (temp1),y
    cpy temp3
    bne goll3
    dec temp4
    bpl goll3
    lda cryptkey+$01
    rts

goll3
    sta cryptkey+$01
    iny
    bne gloop3
    inc temp2
    jmp gloop3

getcrypt4

    ldy #$07
    ldx #$05

copyblock4

    lda blockread,y
```

```
    sta $e2cb,y
    iny
    dex
    bne copyblock4

    lda #$ce
    sta $e46e
    lda #$ee
    sta $e47b
    jsr docrypt4
    dec cryptkey
    lda cryptkey
    cmp cryptkey+$01
    bne gcexit4
    lda #$ee
    sta cryptkey
    inc cryptkey+$01
gcexit4
    rts
docrypt4

    sei
    jsr romsout
    lda #$08
    sta temp1
    lda #$e0
    sta temp2
    lda #$b8
    sta temp3
    lda #$02
    sta temp4
    jsr getkey4
    clc
    adc loadend
    sta cryptkey
    lda #$db
    sta temp1
    lda #$e2
    sta temp2
    lda #$f0
    sta temp3
    lda #$01
    sta temp4
    jsr getkey4
    sta cryptkey+$01
    jsr romsin
    cli
    rts
getkey4
    ldy #$00
```

```
    tya
gccount4
    asl a
    adc (temp1),y
    cpy temp3
    bne goll4
    dec temp4
    bpl goll4
    lda cryptkey+$01
    rts
```

```
goll4
    sta cryptkey+$01
    iny
    bne gccount4
    inc temp2
    jmp gccount4
```

savefile

```
    jsr printsaving
    lda #$01
    sta block
    lda block
    jsr hex2dec
    sty $05a2
    stx $05a3
    sta $05a4
    lda start+$01
    jsr hex2pet
    sty savename+$05
    sta savename+$06
    lda start
    jsr hex2pet
    sty savename+$07
    sta savename+$08
    jsr setstart
    lda #$01
    ldx #$09
    ldy #$01
    jsr setlfs
    lda #$09
    ldx #<savename
    ldy #>savename
    jsr setnam
    jsr open
    ldx #$01
    jsr chkout
    lda #$00
    jsr ciout
    lda loadhigh
```

```
    jsr ciout

    ldy #$00
saveb
    jsr basicout
    inc $d020
    lda (loadaddr),y
    dec $d020
    pha
    jsr basicin
    pla
    jsr ciout
    lda $90
    beq saveok
    pla
    pla
    jmp saveerror
saveok
    inc loadaddr
    bne roll7
    inc loadaddr+$01
    lda block
    jsr hex2dec
    sty $05a2
    stx $05a3
    sta $05a4
    inc block
roll7
    lda loadaddr+$01
    cmp loadend
    bne saveb

    lda #$01
    jsr close
    rts

basicout
    lda $01
    and #$fe
    sta $01
    rts

basicin
    lda $01
    ora #$01
    sta $01
    rts

romsout
```



```
lda $01
and #$fd
sta $01
rts
```

romsin

```
lda $01
ora #$02
sta $01
rts
```

setstart

```
lda #$00
sta loadaddr
lda loadhigh
sta loadaddr+$01
rts
```

fillmem

```
ldx #$b8
ldy #$00
sty loadaddr
lda #$08
sta loadaddr+$01
tya
```

```
floop sta (loadaddr),y
iny
bne floop
inc loadaddr+$01
dex
bne floop
rts
```

loadea

```
lda #$08
tax
ldy #$00
jsr setlfs
lda #$04
ldx #<filename
ldy #>filename
jsr setnam
lda #$00
tax
ldy #$10
sta $9d
jmp load
```

hex2dec

```
    ldy #$2f
    ldx #$3a
    sec

h2dloop1

    iny
    sbc #100
    bcs h2dloop1

h2dloop2

    dex
    adc #10
    bmi h2dloop2
    adc #$2f
    rts

hex2pet

    tax
    lsr
    lsr
    lsr
    lsr
    jsr h2p
    tay
    txa
    and #$0f

h2p
    ora #$30
    cmp #$3a
    bcc h2pexit
    adc #$06

h2pexit
    rts

setupscreen

    lda #$17
    sta $d018
    ldx #$00
    stx $d020
    stx $d021
sloop  lda #$20
    sta $0400,x
    sta $0500,x
    sta $0600,x
    sta $06e7,x
    lda #$01
```

```
sta $d800,x
sta $d900,x
sta $da00,x
sta $dae7,x
inx
bne sloop
```

```
lda #<title
ldx #>title
jsr setorigin
lda # $00
ldx # $04
ldy # $0d
jsr setmessage
lda #<tstatus
ldx #>tstatus
jsr setorigin
lda #<status
ldx #>status
ldy # $07
jmp setmessage
```

printblocks

```
lda #<b2load
ldx #>b2load
jsr setorigin
lda #<blocksinfile
ldx #>blocksinfile
ldy # $01
jmp setmessage
```

printloading

```
lda #<bloading
ldx #>bloading
jsr setorigin
lda #<loadingblock
ldx #>loadingblock
ldy # $01
jmp setmessage
```

printsaving

```
lda #<bsaving
ldx #>bsaving
jsr setorigin
lda #<savingblock
ldx #>savingblock
ldy # $01
jmp setmessage
```

printentry

```
lda #<sjump
ldx #>sjump
jsr setorigin
lda #<entrypoint
ldx #>entrypoint
ldy #$01
jmp setmessage
```

printdone

```
lda #<tdone
ldx #>tdone
jsr setorigin
lda #<statusmessage
ldx #>statusmessage
ldy #$0d
jmp setmessage
```

loaderror

```
lda #<lerror
ldx #>lerror
jsr setorigin
jmp enderror
```

saveerror

```
lda #<serror
ldx #>serror
jsr setorigin
jmp enderror
```

versionerror

```
lda #<verror
ldx #>verror
jsr setorigin
jmp enderror
```

checkerror

```
lda #<cerror
ldx #>cerror
jsr setorigin
jmp enderror
```

enderror

```
lda #<statusmessage
ldx #>statusmessage
```

```
    ldy #$0a
    jsr setmessage
    lda #$02
    sta $d020
    jmp *
```

printversion

```
    lda #<versiondetected
    ldx #>versiondetected
    jsr setorigin
    lda #<versionmessage
    ldx #>versionmessage
    ldy #$0d
    jmp setmessage
```

setorigin

```
    sta temp1
    stx temp2
    rts
```

setmessage

```
    sta temp3
    stx temp4
    sty temp5
    lda temp3
    sta temp6
    lda temp4
    clc
    adc #$d4
    sta temp7
```

```
    ldy #$00
```

smloop

```
    lda (temp1),y
    beq setdone
    bpl roll3
    eor #$80
    bpl roll4
```

roll3

```
    cmp #$3f
    bcc roll4
    eor #$40
```

roll4

```
    sta (temp3),y
    lda temp5
    sta (temp6),y
```

```
    inc temp1
    bne roll5
    inc temp2
```

```
roll5
    inc temp3
    inc temp6
    bne roll6
    inc temp4
    inc temp7
roll6
    jmp smloop

setdone
    rts

;-----
;data

version
    .byte $00

loadhigh
    .byte $00

loadend
    .byte $00

checksum
    .byte $00

cryptkey
    .byte $00,$00

start
    .byte $00,$00

block
    .byte $00

sblocks
    .byte $00

filename
    .byte $45,$41,$22,$9d,$00
```

initialize

```
.text "i0:"  
.byte $00
```

channel

```
.text "#"  
.byte $00
```

blockread

```
.text "u1:2,0,00,00"  
.byte $00
```

bufferpointer

```
.text "b-p:2,0"  
.byte $00
```

savename

```
.text "main      "  
.byte $00
```

b2load

```
.text "Blocks in File :"  
.byte $00
```

bloading

```
.text "Loading Block  :"  
.byte $00
```

bsaving

```
.text "Saving Block   :"  
.byte $00
```

sjump

```
.text "Entry Point   : $"  
.byte $00
```

versiondetected

```
.text "Fat Tracks v0.0 Detected"  
.byte $00
```

tstatus

```
.text "Status  :"  
.byte $00
```

tloading

```
.text "Loading"      "  
.byte $00  
  
tsaving  
  
.text "Saving"      "  
.byte $00  
  
tdone  
  
.text "Done!"      "  
.byte $00  
  
lerror  
  
.text "Load Error!" "  
.byte $00  
  
serror  
  
.text "Save Error!" "  
.byte $00  
  
cerror  
  
.text "Checksum Error! "  
.byte $00  
  
verror  
.text "Unknown Version!"  
.byte $00  
  
;      0123456789012345678901234567890123456789  
title  
.text " Electronic Arts Fat Tracks File Buster "  
.text "                                     "  
.text "Source in Drive 8 Destination in Drive 9"  
.text "                                     "  
.text "           Press SPACE to Transfer!   "  
.byte $00
```

From:
<https://codebase64.org/> - **Codebase 64 wiki**

Permanent link:
https://codebase64.org/doku.php?id=base:electronic_arts_fat_tracks_file_buster

Last update: **2015-04-17 04:31**



