

0 ( COMBLT - VARIABLES  
1 ( COMBLT - ?COLORS HFRAMELEN  
2 ( COMBLT - BLTFRAME  
3 ( COMBLT - PACKHYBRID  
4 ( COMBLT - >TIB  
5 ( COMBLT - SAVEFILE GETSLIDE  
6 ( COMBLT - >RUNBIT  
7 ( COMBLT - >RUNBIT  
8 ( COMBLT - .RUNBIT  
9 ( COMBLT - .RUNBIT  
10 ( COMBLT - .HYBRID  
11 ( COMBLT - measure length of communication blts  
12 ( COMBLT - measure length of communication blts

0

3

```

0 ( COMBLT - VARIABLES
1 EMPTY CR ." NOTICE! EMPTY EXECUTED IN COMBLT.CMP "
2 V= #FRAMES
3 V= FRAMEOFF ( frame offset pointer)
4 ( V= LFRAME )
5 V= BIT-COUNT
6 CREATE FILENAME 15 ALLOT
7 2V= DPVA ( allocation pointer within VESBLT)
8 23040 C= INITIAL-DPB ( start vsa of b disk)
9 CREATE BLTCOLORS 16 ALLOT ( flags showing color exists in blt)
10
11
12
13
14
15

```

```

9-12-85) ( COMBLT - PACKHYBRID
: PACKHYBRID ( xblt yblt lblt wblt seg -- \ make packed blt)
( data record at seg using hybrid packing)
BLTSEG ! WBLT ! LBLT ! YBLT ! XBLT !
>DISPLAY #FRAMES OFF HFRAMELEN 5 FRAMEOFF ! ?COLORS
16 1 DO I 16* I + COLOR ! I BLTCOLORS + C@
IF BLTFRAME LFRAME @ 4 - >R 1 #FRAMES +!
BLTSEG @ FRAMEOFF @ 4 + @DS PAD 1 LCMOVE
PAD PAD I + I >RUNBIT PAD I + @ 2+ I <
IF @DS PAD I + DUP @ 2+ >R BLTSEG @ FRAMEOFF @
4 + I LCMOVE
I 4 + BLTSEG @ FRAMEOFF @ L! ( adjust framelen)
1 BLTSEG @ FRAMEOFF @ 3 + LC! R> R> DROP >R
THEN R> 4 + FRAMEOFF +! THEN LOOP BLTSEG @ >R
FRAMEOFF @ I 0 L! LBLT @ I 2 LC! WBLT @ I 3 LC!
#FRAMES @ R> 4 LC! ;

```

1

4

```

0 ( COMBLT - ?COLORS HFRAMELEN
1
2 : ?COLORS ( -- \ set blt color flags based on contents of blt)
3 BLTCOLORS 16 0 FILL
4 LBLT @ 0 DO
5 WBLT @ 0 DO YBLT @ J - XBLT @ I + L@PIXEL
6 BLTCOLORS + 1 SWAP C!
7 LOOP LOOP ;
8
9
10 : HFRAMELEN ( -- \ compute the length of a hybrid blt frame)
11 ( [framelen][color][image/run][image] )
12 LBLT @ WBLT @ * 15 + 16/ 2* 4 + LFRAME ! ;
13
14
15

```

```

9-12-85) ( COMBLT - >TIB
: >TIB ( addr count -- \ move the string referred to by the )
( stack to the terminal input buffer and append a null)
>IN OFF 2DUP TIB @ SWAP CMOVE
TIB @ + 0 SWAP ! DROP ;

```

2

5

```

0 ( COMBLT - BLTFRAME
1 : BLTFRAME ( -- \ build a blt image for the current color)
2 ( relative to bltseg & frameoff)
3 BIT-COUNT OFF FRAMEOFF @ DUP >R 4 + ABLT ! BLTSEG @ >R
4 LFRAME @ I I' L! COLOR @ I I' 2+ LC! 0 R> R> 3+ LC!
5 0
6 LBLT @ 0 DO
7 WBLT @ 0 DO YBLT @ J - XBLT @ I + L@PIXEL
8 COLOR @ 15 AND = SHL-BIT 1 BIT-COUNT +!
9 BIT-COUNT @ 16 =
10 IF BLTSEG @ ABLT @ L! 2 ABLT +! 0. BIT-COUNT !
11 THEN
12 LOOP LOOP
13 BIT-COUNT @ ?DUP IF 16 SWAP DO 2* LOOP BLTSEG @ ABLT @ L!
14 ELSE DROP THEN ;
15

```

```

9-12-85) ( COMBLT - SAVEFILE GETSLIDE
: SAVEFILE ( -- \ save current dos file name)
FILENAME 15 BL FILL
FILENAME
8 0 DO I DOS NAM + C@ DUP BL =
IF DROP LEAVE ELSE OVER C! 1+ THEN LOOP
ASCII . OVER C! 1+
3 0 DO I DOS TYP + C@ DUP BL =
IF DROP LEAVE ELSE OVER C! 1+ THEN LOOP DROP ;
: GETSLIDE ( addr count -- \ get slide in file referred to by)
( the stack, send it to the display buffer and reopen the )
( current file)
>IN @ >R BLK @ >R SAVEFILE >LORES >DISPLAY
>TIB BLK OFF USING 0 BLK>BUFFER
FILENAME 15 >TIB USING R> BLK ! R> >IN ! ;

```

6

9

```

0 ( COMBLT - >RUNBIT
1 EXIT HEX
2 CODE >RUNBIT ( source dest len -- \ bit wise run length )
3 ( encoder using word size fetching; BLT compatible)
4 1 POP 1 SHL 1 SHL 1 SHL 0 POP 2 POP W PUSH I PUSH R PUSH
5 0 PUSH 0 INC 0 INC 0 W MOV 2 I MOV 100 # 2 MOV U U XOR
6 R R XOR BEGIN U U OR B 0= IF LODS 10 # U MOV
7 THEN U HI U HI XOR B 0 RCL U HI RCL B FE C, CB C,
8 U HI 2 HI CMP B
9 0= IF FE C, C2 C, ELSE 2 W ) MOV B W INC R INC 2 HI 2 HI OR B
10 0= IF FE C, C6 C, ELSE FE C, CE C, THEN 1 #B 2 MOV THEN
11 FF #B 2 CMP 0= IF 2 W ) MOV B W INC R INC 2 2 XOR B
12 2 HI 2 HI OR B 0= IF FE C, C6 C, ELSE FE C, CE C, THEN THEN
13 LOOP 2 W ) MOV B R INC W POP R W ) MOV R POP I POP
14 W POP NEXT
15 DECIMAL

```

9-12-85) ( COMBLT - .RUNBIT

9-12-85)

```

HEX
X1 , 3B C, 06 C, Y2 , 75 C, 0C C, FF C, 36 C, XBLT , 8F C,
06 C, X1 , FF C, 0E C, Y1 , E2 C, B5 C, EB C, 1E C, 01 C,
0E C, X1 , 8B C, 06 C, X1 , 3B C, 06 C, Y2 , 7B C, 0C C,
2B C, 06 C, WBLT , FF C, 0E C, Y1 , EB C, F0 C, EB C, 04 C,
89 C, 06 C, X1 , 59 C, 49 C, 74 C, 03 C, E9 C, 6D C, FF C,
5F C, 5E C, 1E C, 07 C, FB C, AD C, 8B C, DB C, FF C, 27 C,
DECIMAL

```

BEFORE

7

10

```

0 ( COMBLT - >RUNBIT
1 HEX
2 CREATE >RUNBIT ( source dest len -- \ bit wise run length )
3 ( encoder using word size fetching; BLT compatible)
4 HERE DUP 2- ! 59 C, D1 C, E1 C, D1 C, E1 C, D1 C, E1 C,
5 5B C, 5A C, 57 C, 56 C, 55 C, 50 C, 40 C, 40 C, 8B C, FB C,
6 8B C, F2 C, 8A C, 00 C, 01 C, 33 C, DB C, 33 C, ED C, 0A C,
7 DB C, 75 C, 04 C, AD C, 8B C, 10 C, 00 C, 32 C, FF C, D1 C,
8 D0 C, D0 C, D7 C, FE C, CB C, 3A C, F7 C, 75 C, 04 C, FE C,
9 C2 C, EB C, 10 C, 8B C, 15 C, 47 C, 45 C, 0A C, F6 C, 75 C,
10 04 C, FE C, C6 C, EB C, 02 C, FE C, CE C, B2 C, 01 C, 80 C,
11 FA C, FF C, 75 C, 10 C, 8B C, 15 C, 47 C, 45 C, 32 C, D2 C,
12 0A C, F6 C, 75 C, 04 C, FE C, C6 C, EB C, 02 C, FE C, CE C,
13 E2 C, C1 C, 8B C, 15 C, 45 C, 5F C, 89 C, 2D C, 5D C, 5E C,
14 5F C, AD C, 8B C, DB C, FF C, 27 C,
15 DECIMAL

```

9-12-85) ( COMBLT - .HYBRID

9-12-85)

```

: .HYBRID ( seg -- \ plot a hybrid packed blt at seg; assumes)
( xblt, yblt are set)
DUP BLTSEG ! >R 5 ABLT !
1 2 L@ LBLT ! 1 3 L@ WBLT !
R> 4 L@ 0 DO ABLT @ >R BLTSEG @ >R 4 ABLT +!
I I' 2+ L@ COLOR ! R> 3 + L@
IF BLTSEG @ ABLT @ .RUNBIT
ELSE (BLT)
THEN
BLTSEG @ ABLT @ 4 - L@ 4 - ABLT +!
LOOP ;

```

8

11

```

0 ( COMBLT - .RUNBIT
1 HEX
2 CREATE .RUNBIT ( seg off -- \ given location of run encoded)
3 ( string and XBLT, YBLT, WBLT, LBLT and COLOR set, plot BLT)
4 HERE DUP 2- ! FA C, FF C, 36 C, XBLT , 8F C, 06 C, X1 ,
5 FF C, 36 C, YBLT , 8F C, 06 C, Y1 , 5B C, 07 C, 26 C, 8B C,
6 0F C, 06 C, 8F C, 06 C, BLTSEG , 56 C, 57 C, 83 C, C3 C, 02 C,
7 89 C, 1E C, X2 , 8B C, 06 C, XBLT , 03 C, 06 C, WBLT , 89 C,
8 06 C, Y2 , 51 C, 33 C, C9 C, FF C, 36 C, BLTSEG , 07 C, 26 C,
9 8A C, 0F C, 43 C, 0B C, C9 C, 74 C, 7C C, 8B C, C3 C, 4B C,
10 2B C, 06 C, X2 , 25 C, 01 C, 00 C, 75 C, 52 C, FF C, 36 C,
11 BUF-SEG , 07 C, 8B C, 16 C, Y1 , D1 C, E2 C, 8B C, 06 C, X1 ,
12 03 C, 16 C, YTABL , 8B C, FA C, D1 C, EB C, 8A C, F0 C, 00 C,
13 73 C, 02 C, B2 C, 0F C, 03 C, 05 C, 8B C, FB C, 26 C, 8A C,
14 05 C, 8A C, 36 C, COLOR , 22 C, F2 C, F6 C, D2 C, 22 C, C2 C,
15 0A C, C6 C, 26 C, 8B C, 05 C, FF C, 06 C, X1 , 8B C, 06 C,

```

9-12-85) ( COMBLT - measure length of communication blts

9-12-85)

```

EXIT
: LENRUN ( x y l w addr cnt -- \ )
( given file name string compress )
( ship blt and display length of compress string)
>TIB USING >LORES >DISPLAY 0 BLK>BUFFER
HBUF-SEG @ PACKHYBRID DARK
HBUF-SEG @ .HYBRID KEY DROP
>ALPHA HBUF-SEG @ 0 L@ U. KEY DROP ;
: test 0 99 100 160 " STARPORT.PIC" LENRUN
0 199 100 160 " STARPORT.PIC" LENRUN
4 191 120 72 " ELOWAN.PIC" lenrun
4 191 120 72 " SPEMIN.PIC" lenrun
4 191 120 72 " THRYNN.PIC" lenrun
4 191 120 72 " VELOX.PIC" lenrun ;

```

12

```
0 ( COMBLT - measure length of communication blts      9-10-85)
1 exit
2 : test
3      2 194 120 72 " MYSTER.PIC" lenrun
4      4 191 120 72 " GAZURT.PIC" lenrun
5      4 191 120 72 " MECHAN.PIC" lenrun
6      4 191 120 72 " NOMAD.PIC" lenrun
7      4 191 120 72 " VPROBE.PIC" lenrun
8      4 191 120 72 " MINSTREL.PIC" lenrun ;
9
10
11
12
13
14
15
```

13

```
0 ( COMBLT - CMPBLT3      9-12-85)
1
2 : CMPBLT3 ( dir# x y l w addr cnt -- \ given file name string)
3 ( compress communications image and save on disk)
4 >TIB USING >LORES >DISPLAY 0 BLK>BUFFER
5 HBUF-SEG @ PACKHYBRID DARK
6 HBUF-SEG @ .HYBRID HBUF-SEG @ SWAP >FILE >ALPHA ;
7
8 6 4 191 120 72 " ELOWAN.PIC" CMPBLT3
9 31 4 191 120 72 " SPEMIN.PIC" CMPBLT3
10 33 4 191 120 72 " THRYNN.PIC" CMPBLT3
11 37 4 191 120 72 " VELOX.PIC" CMPBLT3
12 44 0 199 100 160 " STARPORT.PIC" CMPBLT3
13 49 0 99 100 160 " STARPORT.PIC" CMPBLT3
14
15 FORGET #FRAMES
```