

"REPACK" BY MICHAEL WHITE

ARCADIAN VOL.6 PG.108 AND 109 SHOWS THE "OLD" LISTING AND PROCEDURE FOR THIS UTILITY! HOWEVER, IF YOU CHANGE FILE SEARCH SYSTEMS TO THE ":PRINT H,18" METHOD, OTHER THINGS CHANGE ALSO! THE COMPLETE PROCEDURE FOR THE "NO ADDED MEMORY" REPACKING (USING A ":PRINT H,18" FILING) IS:

- .1.KEY IN OBJECT PROGRAM TEXT, VARIABLES, STRINGS, AND MACHINE CODE IN STACK AREA
- .2.AFTER DEBUGGING, TAPE MACHINE CODE WITH :PRINT %(20237),73 [OR WHATEVER] (ON A WORK TAPE)
- .3.RUN 30 SECONDS OF BLANK TAPE THROUGH AND TAPE THE TEXT AND VARIABLES THERE (ON SAME WORK TAPE) BY USING :PRINT %(-24576),904;:PRINT %(20000),57
- .4.REWIND WORK TAPE, ENTER :LIST [GO], AND PLAY TAPE STOPPING JUST WHEN THE CURSOR RETURNS
- .5.PRESS [RESET] AND KEY IN YOUR PICTURE PROGRAM OR LOAD YOUR FAVORITE "DOODLE" PROGRAM AND DRAW YOUR PICTURE. IF EB IS USED TO CREATE YOUR PICTURE, ONLY THE FB (AND OR FC) COLORS CAN BE USED AS "REPACK" DISCARDS THE ODD (FA) BITS WHEN IT REPLACES THEM WITH THE OBJECT PROGRAM'S BASIC TEXT
- .6.TAPE PICTURE PROGRAM INSIDE THE 30 SECONDS OF BLANK TAPE CREATED IN STEP 3 WITH THE PICTURE SHOWING! NOTE: USE :PRINT %(16384),1864
- .7.SWITCH TO YOUR FINISHED FILED TAPE AND CUE FOR THE NEXT "SLAVE" PROGRAM
- .8.KEY THE "BURST" VALUES INTO THE VARIABLES (ALL 18 IF NECESSARY)
- .9.KEY IN ":PRINT H,18" START THE TAPE RECORDING, WAIT 5 SECONDS AND HIT [GO]
- .10.VERIFY YOUR "BURST" WITH :LIST
- .11.SWITCH BACK TO THE WORK TAPE, REWIND, AND RELOAD THE MACHINE CODE WITH :INPUT %(20237) [OR WHATEVER]
- .12.ENTER THIS LINE, PRESS [GO] AND RESUME PLAYING THE WORK TAPE (LEAVE THE TAPE RUNNING UNTIL YOU'RE SURE IT'S ALL LOADED IN!)

```
PRINT ";RUN ";:INPUT ;:INPUT %(-24576);:INPUT %(20000);  
IF KP:PRINT %(16384),2000
```

- .13.SWITCH BACK TO YOUR FINISHED FILED TAPE AND CUE IT JUST BEHIND THE "BURST" THAT YOU TAPED IN STEP 9
- .14.START TAPE RECORDING, WAIT 2 SECONDS AND HIT A KEY
- .15.VERIFY WITH THE :LIST FUNCTION
- .16.REWIND YOUR FINISHED FILED TAPE AND LOAD THE MENU PROGRAM WITH :INPUT [GO]
- .17.WHEN THE MENU APPEARS, PRESS [HALT] AND LIST TO FIND THE (LAST LINE NUMBER)
- .18.USING THAT LINE NUMBER (IN PLACE OF THE "XX0" SHOWN BELOW) KEY IN:

```
>XX0 IF G=(CODE NUMBER)CX=-(# OF CHARACTORS AND SPACES-1)x3;PRINT "(NEW TITLE)
```

- .19.FINISH WITH THE KEY SEQUENCE [WORDS] [GO] [WORDS] RETURN [GO]
- .20.SWITCH TO YOUR WORK TAPE, REWIND, AND SAVE (USING THE TAPING COMMAND IN STEP 3)
- .21.REWIND THE WORK TAPE, SWITCH TO YOUR FINISHED FILED TAPE AND ALSO REWIND
- .22.KEY IN THE LINE SHOWN IN STEP 12 (CHANGING THE "2000" TO "1887"), HIT [GO] AND PLAY YOUR OLD MENU PROGRAM BACK IN, THEN SWITCH BACK TO YOUR WORK TAPE AND PLAY THE NEW TEXT IN ALSO
- .23.SWITCH TO YOUR FINISHED FILED TAPE, REWIND, START THE TAPE RECORDING AND HIT [GO] (SO AS TO TAPE OVER THE OLD MENU PROGRAM)
- .24.VERIFY THE MENU PROGRAM WITH :LIST , THEN REWIND ONE MORE TIME AND TRY IT!!

THE ABOVE PROCEDURE WILL GET A NEW "SLAVE" PROGRAM ADDED TO YOUR "MASTER" FILE SEARCHED TAPE, AND UPDATE THE MENU, BUT REQUIRES QUITE A BIT OF TAPE SHUFFLING! IF YOU HAVE A MEMORY EXPANSION (ANY WILL DO THAT HAS AT LEAST 8K OF RAM), THEN KEY IN THE FOLLOWING (AB) LISTING, AND SAVE TO A "UTILITIES" TAPE:

```

>10 CLEAR ;CY=10;CX=-21;PRINT "ZEROING";CX=-9;PRINT "RAM";CX=-18;PRINT "PLEASE"
;CX=-12;PRINT "WAIT
>20 A=24576;B=20018;C=-A;D=-22772;E=19998;F=20384;G=16384;H=28600;I=30450;
J=30850;K=30900
>30 L=%(20000);M=8192;N=2258;O=32766;FOR P=ATO OSTEP 2;%(P)=0;NEXT P;CLEAR ;
PRINT " READY
>40 PRINT " Awaiting INPUT ";PRINT " OF TEXT";:INPUT %(A);
PRINT " TEXT LOADED!
>50 R=K;FOR P=CTO DSTEP 2;%(R)=%(P);R=R+2;NEXT P;R=H;PRINT " UNPACKING TEXT!
>60 FOR P=GTO ESTEP 2;%(P)=%(P+M);NEXT P;FOR P=CTO DSTEP 2;%(R)=%(P);R=R+2;
NEXT P;R=K;FOR P=CTO LSTEP 2;%(P)=%(R);R=R+2;NEXT P
>70 CLEAR ;PRINT " TEXT UNPACKED";FOR P=ITO JSTEP 2;%(P)=%(P-N);NEXT P;
PRINT " READY
>80 PRINT " TO LOAD PICTURE";:INPUT %(A);PRINT " PICTURE LOADED!
>90 FOR P=0TO N;NEXT P;PRINT " REPACKING TEXT!
>100 FOR P=GTO ESTEP 2;%(P)=%(P+M);NEXT P;R=K;FOR P=CTO LSTEP 2;%(P)=%(R);
R=R+2;NEXT P;R=H
>110 FOR P=CTO DSTEP 2;%(P)=%(R);R=R+2;NEXT P;FOR P=GTO ESTEP 2;%(P+M)=%(P);
NEXT P;R=K;FOR P=CTO LSTEP 2;%(P)=%(R);R=R+2;NEXT P
>120 CLEAR ;PRINT " TEXT REPACKED";FOR P=ITO JSTEP 2;%(P-N)=%(P);IF %(P)
C=(P-A-N)/2
>130 NEXT P;PRINT " DO YOU WANT";PRINT " FILE SEARCH?";GOSUB 210
>140 D=KP-50;IF (D<-1)+(D>0)GOTO 140
>150 CLEAR ;F=B;IF DFOR G=72TO 89;TV=32;TV=G;INPUT "="%(F);F=F+2;NEXT G;
PRINT " IS THIS OK?";GOSUB 210;IF KP-49GOTO 150
>160 CLEAR ;PRINT " START TAPE RECORDING";PRINT " AND HIT ANY KEY
>170 IF KPIF D:PRINT H,18
>180 :PRINT %(A),C;PRINT " :LIST ";:LIST ;IF D:LIST
>190 PRINT ;PRINT " TO RERUN UTILITY";PRINT " PRESS [G0]";IF KP=13RUN
>200 STOP
>210 PRINT " 1=YES 2=NO";RETURN

```

USE: >PRINT ";RUN ";:PRINT %(16384),1887 [G0] (TO SAVE TO TAPE)

TO ASSEMBLE A FINISHED "SLAVE" PROGRAM, FOLLOW THE INSTRUCTIONS ON PG.108 OF ARCADIAN VOL.6 (AND THE ON SCREEN PROMPTS)! THEN UPDATE YOUR "MENU" PROGRAM BY EMPLOYING STEPS 16 THROUGH 19 ABOVE, AND THEN "REPACK" IT WITH THE OLD MENU PROGRAM'S TITLE SCREEN.

WHAT EXACTLY DOES "REPACK" DO? (SOME OUT THERE MAY ASK). "REPACK" CHANGES THE "TITLE SCREEN" ON A PREVIOUSLY TAPED PROGRAM (AB OR BB), MEANING THAT THE "TITLE SCREEN" (WHICH APPEARS WHILE THE TAPE IS LOADING) CAN NOW BE CREATED ELSEWHERE! NOTE: THIS PROGRAM WAS ORIGINALLY WRITTEN TO PUT "TITLE SCREENS" ON OLD BB (300 BAUD) PROGRAMS! (MORE ON THAT LATER)!

THE "SECRET" TO "REPACK" IS IN THE FACT THAT AB (AND BB) RECOGNISE THEIR "TEXT AREA" BY THE PHANTOM ADDRESSES %(-24576) THROUGH %(-22772), WHICH MEANS THAT ALTHOUGH WE CAN'T CHANGE A "TITLE SCREEN", "PER SE", WE CAN CHANGE TEXTS!! HOWEVER, THIS ONLY WORKS IN THE SCREEN AREA, AND ONLY IF AB OR BB IS IN THE CARTRIDGE SLOT! WHICH LEAVES NO ROOM FOR OUR UTILITY IN AB OR BB AND LEAVES OUT EB, ENTIRELY! (ALMOST! MORE ON THIS LATER).

THESE DIFFICULTIES ARE GOTTEN AROUND BY "TIME SHARING" IN AB, DUE TO THE FACT THAT AB (AND ONLY AB!) COPIES EACH LINE OF BASIC TEXT INTO THE LINE INPUT BUFFER, AND EXECUTES IT FROM THERE! AS LONG AS IT DOESN'T NEED TO "FETCH" ANOTHER LINE FOR ANY REASON, IT STAYS IN THE LINE INPUT BUFFER!

THE ABOVE LISTING USES 8K OF EXTENDED MEMORY %(24576) THROUGH %(32767) WHICH IS AVAILABLE IN EVERY MEMORY EXPANSION SCHEME (LARGER THAN 4K) THAT IS CURRENTLY KNOWN! THE UTILITY BEGINS (IN LINES 10, 20, AND 30) BY ZEROING OUT THE ENTIRE AREA! THIS IS SO IT CAN LOCATE THE END OF THE OBJECT PROGRAM'S TEXT FILE AND DOESN'T GET CONFUSED!

IN LINES 40 AND 50 THE OBJECT PROGRAM IS LOADED INTO EXTENDED MEMORY AND THE UTILITY "COPIES" ITS OWN TEXT THERE ALSO. (USING "PEEK AND POKE" LOOPS MAKES IT POSSIBLE TO SEE WHAT'S HAPPENING IN BASIC!) NOTE: THE UTILITY'S TEXT

IS 1800 BYTES (OR LESS), AND SO IS THE TEXT OF OUR OBJECT PROGRAM! THAT'S 3600 OUT OF 4096, WHICH LEAVES PLENTY OF ROOM TO COPY THE OBJECT PROGRAM'S "SCRATCHPAD" [%(20000) THROUGH %(20384)] INSIDE THE SAME (2ND) 4K, WHILE OUR TAPE LOADING AND SAVING IS BEING DONE WITH THE OTHER (1ST) 4K, (HENSEFORTH THE NEED FOR 8K)! THE ADDRESSES WHERE ALL THIS IS KEPT IS DETERMINED IN LINES 20 AND 30.

LINE 60 "UNPACKS" THE OBJECT PROGRAM'S TEXT! THE FIRST LOOP MOVES THE TEXT PROGRAM TO THE SCREEN (YOU CAN SEE THIS HAPPENING)! THE SECOND LOOP "UNPACKS" THE OBJECT PROGRAM TEXT TO THE 2ND 4K, AND THE THIRD LOOP "RESCUES" THE TEXT OF THE UTILITY BEFORE THE COMPUTER HAS TO "FETCH" LINE 70! **WARNING!!!** IF YOU PRESS [HALT] WHILE THIS LINE IS EXECUTING ALL IS LOST!!! NOTICE THE MESSAGES THAT COME UP IN LINES 50 AND 70! **DON'T [HALT] IN BETWEEN!!** LINE 70 ALSO MOVES THE OBJECT PROGRAM'S "SCRATCHPAD" FROM THE 1ST 4K AREA TO THE 2ND 4K, SO THAT THE PICTURE PROGRAM CAN BE LOADED INTO THE 1ST 4K IN LINE 80.

LINE 90 RUNS A SHORT TIME DELAY LOOP SO YOU CAN [HALT] AND KEY IN: GOTO 80 IF YOU GET A BAD TAPE LOAD, BUT BE QUICK!! AFTER THE "REPACKING TEXT" MESSAGE APPEARS, THE UTILITY "DISAPPEARS" AGAIN (SAME AS LINE 60) UNTIL THE MESSAGE "TEXT REPACKED" PRINTS OUT FROM LINE 120.

AT THIS POINT ANOTHER DILEMMA WAS ENCOUNTERED! THE 102 BYTE LINE INPUT BUFFER ISN'T BIG ENOUGH TO HOLD 4 LOOPS, SO WE USE 2 LINES AND 5 LOOPS! THE FIRST LOOP (LINE 100) MOVES THE PICTURE PROGRAM TO THE SCREEN FROM THE 1ST 4K. (THIS PART YOU CAN WATCH), THEN THE SECOND LOOP RE-INSTATES THE UTILITY SO THAT LINE 110 CAN BE "FETCHED" TO DO THE REST! THE THIRD LOOP (FIRST IN LINE 110) "REPACKS" THE OBJECT PROGRAM TEXT TO THE SCREEN! THE FOURTH LOOP (SECOND IN LINE 110) MOVES THE COMPOSITE TO THE 1ST 4K FROM THE SCREEN, AND THE LAST LOOP (IN LINE 110) DOES THE SAME THING AS THE SECOND LOOP IN LINE 100, OR THE THIRD LOOP IN LINE 60, SO THE UTILITY CAN CONTINUE!

LINE 120 RUNS A LOOP THAT MOVES THE "SCRATCHPAD" OF OUR OBJECT PROGRAM BACK TO THE 1ST 4K, AND SEARCHES FOR THE END OF THE OBJECT PROGRAM AT THE SAME TIME BY LOOKING FOR BYTES THAT CONTAIN ANYTHING BUT ZERO "0" (WHICH IS WHAT LINE 30 "STARTED" US OUT WITH)! THE LAST NON-ZERO BYTE SETS VARIABLE "C" TO THE NUMBER OF WORDS THAT THE TAPING COMMAND (IN LINE 180) NEEDS!

LINE 130 ASKS IF YOU WANT FILE SEARCH, AND LINE 140 AWAITS YOUR DECISION. YOU HAVE THREE CHOICES, YES, NO, AND [HALT]! IF YOU SAY "YES" LINE 150 SETS UP THE INPUTS FOR A "BURST" THAT PUTS ALL 18 VARIABLES ON THE TAPE BY "POKING" INTO THE VARIABLE'S ADDRESSES (SEE ARCADIAN VOL.5 PG.59). AND THEN IT GETS SENT TO YOUR FINISHED FILED "MASTER" TAPE IN LINE 170! IF YOU SAY "NO" THIS PART IS BYPASSED SOMEWHAT, AND THE COMPOSITE PROGRAM IS SAVED TO TAPE WITHOUT THE 18 VARIABLE "BURST". IN EITHER CASE LINE 180 AUTOMATICALLY VERIFIES THE TAPE, AND IF IT RECORDED BAD, JUST PRESS [HALT] AND KEY IN: GOTO 160 [GO] AND TRY AGAIN.

HOWEVER, IF YOU CHOOSE [HALT] AT LINE 140, SWITCH YOUR EXTENDED MEMORY TO THE 6K ROM MODE (IF YOU HAVE THAT ABILITY), AND IF NOT JUST HOLD THE [RESET] BUTTON IN WHILE YOU SWAP CARTRIDGES TO OLD BALLY BASIC! IF YOU HAVE THE 300 BAUD TAPING INTERFACE CONNECTED, YOU CAN NOW KEY IN (OR LOAD IN AT 300 BAUD) A SHORT "DUMPING" PROGRAM THAT PUTS THE OBJECT PROGRAM ONTO THE TAPE WITH THE TITLE SCREEN, AT 300 BAUD! THIS USES 5 MINUTES OF TAPE AND DOESN'T SAVE THE VARIABLES, OR ANYTHING ELSE, (INCLUDING THE ARCADIAN VOL.3 PG.4 FILE SEARCH), UNLESS YOU INCLUDE IT IN YOUR "DUMPING" PROGRAM, WHICH SHOULD RESEMBLE:

```
>10 PRINT ;PRINT "IF G#",#1,G,"GOTO 70";FOR F=1TO 999;NEXT F
>20 TV=0;TV=G;TV=13;FOR F=1TO 2000;NEXT F;CLEAR
>30 A=28192;B=%(A);C=A+4;D=A+52;E=20078;NT=5
>40 FOR F=C TO D STEP 2;E=E+2;PRINT "%(",#5,E,")=",%(F);NEXT F;FOR F=1TO 999;
NEXT F;NT=1
>50 PRINT "&(0)=(?);&(1)=(?);&(2)=(?);&(3)=(?);&(9)=(?);&(10)=(?);(ETC.);";
FOR F=1TO 999;NEXT F
>60 PRINT "CLEAR ;FOR A=16384TO 19999;%(A)=KP;NEXT A;%(20050)=",#6,B,"";
A=",#1,%(C-2),";:RETURN ;RUN
>70 NT=0;FOR F=24576TO 28191;CY=0;TV=%(F);NEXT F;:RETURN
```

THEN KEY IN: NT=1;CLEAR ;:PRINT ;G=(YOUR FILE CODE NO.) [GO]
KEY IN: RUN (GET TAPE STARTED AND HIT [GO] AGAIN!

LINES 30, 40, AND 50 OF THIS "DUMPING" PROGRAM ARE NOT NECESSARY AND COULD BE OMITTED OR CHANGED TO OTHER ROUTINES THAT DO THE SAME, OR WHATEVER YOU WISH, LIKE PUT A MACHINE CODE ROUTINE INTO YOUR OBJECT PROGRAM FOR INSTANCE. NOTE: THE LOOP IN LINE 40 PUTS THE VARIABLES (AS THEY WERE SAVED IN THE AB COMPOSITE PROGRAM, NOW RESIDING IN THE 1ST 4K) (ALL EXCEPT "A") ONTO THE TAPE! THE LOADING INSTRUCTION USES "A" AND RESETS IT (FROM AB'S "A") IN LINE 60! ALSO, THE VARIABLE "G" STORES YOUR FILE SEARCH CODE AND MUST BE INPUTTED BY HAND. IF ANYONE WANTS TO SEE THE MENU PROGRAM FOR OLD BB, YOU'LL HAVE TO ASK ME. I DON'T KNOW WHAT INTEREST IS LEFT OUT THERE, BUT THERE ARE A FEW SURPRISES WITH IT!

ANOTHER WAY TO "REPACK" AB OR BB PROGRAMS (EB PROGRAMS DON'T NEED IT) IS TO USE MACHINE CODE ROUTINES! THIS ALLOWS ONE TO USE EB TO "REPACK" WITH, BUT BE SURE YOU HAVE 8K OF UNUSED RAM TO PUT YOUR OBJECT PROGRAM'S PARTS INTO! THE ADVANTAGES TO USING EB TO "REPACK" WITH ARE: 1 DATA STATEMENTS SAVE BYTES (SEE LINES 20 AND 30 OF AB LISTING ABOVE)! 2 THE LINE INPUT BUFFER HAS 160 BYTES INSTEAD OF 102 (ALSO SEE ABOVE). 3 EB ACCESSES THE 300 BAUD INTERFACE (MEANING THAT OUR "DUMPING" PROGRAM AND "REPACK" CAN NOW BE COMBINED INTO ONE PROGRAM). AND 4 THE "LC" VARIABLE OF EB COULD BE EMPLOYED (THIS WAS USED IN SNOOP CAMERA TO CHECK LOADINGS OF TAPED SCREENS, SEE ARCADIAN VOL.7 PG.63 LINE 71 AND PG.65 LINE 5)! "LC" IS A TWO LETTER VARIABLE THAT STANDS FOR "LAST CHARACTER" (THAT GOT PRINTED). LC=63 (NOT ZERO "0") IF THE LAST CHARACTER PRINTED WAS A QUESTION MARK "?", SUCH AS WHAT OCCURS WHEN THE CHECKSUM BYTE DOESN'T CHECK OUT! SO OUR BASIC PROGRAM HAS A WAY OF DETECTING A BAD LOAD OR RECORDING!

THE TWO "ON BOARD" MACHINE CODE ROUTINES THAT COULD BE USED IN "REPACK" (AB VERSION) ARE FILL AND MOVE. FILL REPLACES THE "ZEROING RAM" LOOP IN LINE 30 WITH A CALL ROUTINE THAT RUNS LESS THAN A SECOND! MOVE REPLACES THE FIRST LOOP IN LINE 60, THE LOOP IN LINE 70, THE FIRST LOOP IN LINE 100, THE SECOND LOOP IN LINE 110 (RESULTING IN THESE TWO LINES BECOMING ONE, AND ELIMINATING THE NEED FOR THE SECOND LOOP OF LINE 100), AND FINALLY, THE LOOP IN LINE 120 (EXCEPT THE "SEARCH" FUNCTION). THE FORMAT WE USE TO CALL THESE ROUTINES FROM BASIC IS SIMILAR TO "CHRDIS" (SEE ARCADIAN VOL.5 PG.14)! THAT IS: 1 SAVE DE REGISTER, 2 CALL ROUTINE, 3 RESTORE DE, AND 4 RETURN TO BASIC! THE FILL ROUTINE IS:

	DEC	HEX	ASSY	NOTES
FILL	213	D5	PUSH DE	;SAVE BASIC TEXT PROGRAM POINTER
	255	FF	RST 38	;CALL "ON BOARD" ROUTINE
	27	1B	DEFB	;FILL SUBROUTINE IDENTIFIER (+1)
	24576	00	DEFW	;STARTING ADDRESS [DE]
		60		;(THIS ADDRESS IS !9000 IN EB, !8000 IN BRB)
	8192	00	DEFW	;NO. OF BYTES TO FILL [BC]
		20		
	00	00	DEFB	;VALUE TO FILL BLOCK WITH [A]
	209	D1	POP DE	;RESTORE BASIC TEXT PROGRAM POINTER
	201	C9	RET	;RETURN (TO BASIC)

NOTICE THE SIMILARITIES TO THE MOVE ROUTINE, WHICH IS:

	DEC	HEX	ASSY	NOTES
MOVE	213	D5	PUSH DE	;SAME AS ABOVE
	255	FF	RST 38	;SAME ALSO
	95	5F	DEFB	;MOVE SUBROUTINE IDENTIFIER (+1)
	16384	00	DEFW	;"DESTINATION" ADDRESS [DE]
		40		
	3608	0C	DEFW	;NO. OF "BYTES TO MOVE" [BC]
		07		
	24576	00	DEFW	;"SCORCE" ADDRESS [HL]
		60		
	209	D1	POP DE	;SAME AS ABOVE AGAIN
	201	C9	RET	;DITTO

THE FILL ROUTINE SIMPLY "FILLS" A "BLOCK" OF MEMORY WITH A CONSTANT VALUE, IN THIS CASE ZERO "00"! THE SUBROUTINE IS ACTUALLY DECIMAL NO.26 BUT WE USE 27 SO AS TO ACTIVATE THE "SYSSUK" FUNCTION OF "RST 38", OTHERWISE WE WOULD HAVE TO PRE-LOAD REGISTERS [DE], [BC], AND [A] OURSELVES! THE SAME IS TRUE FOR NO.94 (MOVE), EXCEPT REGISTER [HL] WOULD ALSO NEED PRE-LOADED (BUT NOT [A]). NOTE: THE "MOVE" LISTING (ABOVE) SHOWS WHAT IS REQUIRED TO REPLACE THE 1ST LOOP OF LINE 60. TO REPLACE THE OTHER LOOPS WE MUST POKE "%(X)" INTO THE "DESTINATION", "BYTES TO MOVE", AND "SCORCE", SO AS TO CHANGE THE SIZE OR DIRECTION OF THE MOVE, USING THE METHOD "CHRDIS" USED TO CHANGE CHARACTORS OR LOCATIONS (SEE ARCADIAN VOL.5 PG.37).

TO "REPACK" THE TEXT IN EB THE "ON BOARD" (RST 38) SUBROUTINES CAN'T HELP YOU! YOU NEED A "HOMEBUILT" ROUTINE, SUCH AS:

```

      HEX ASSY      NOTES
REPACK D9 EXX      ;ANOTHER WAY TO SAVE [DE]
      01 LD BC,E18 ;NO. OF BYTES TO "REPACK"
      18
      0E
      11 LD DE,A000;ADDRESS OF PICTURE (USE !8000 FOR BRB)
      00
      A0
      21 LD HL,9000;ADDRESS OF TEXT
      00
      90
LOOP  7E LD A,(HL) ;GET TEXT
      E6 AND 55    ;MASK OUT PICTURE
      55
      77 LD (HL),A ;PUT TEXT BACK WITH A CLEAR SCREEN!
      1A LD A,(DE) ;GET PICTURE
      E6 AND AA    ;MASK OUT TEXT
      AA
      B6 OR (HL)   ;"REPACK" OBJECT PROGRAM (CREATE COMPOSITE)
      77 LD (HL),A ;PUT COMPOSITE IN !9000 AREA (1ST 4K)
      ED LDI      ;DO "HOUSEKEEPING"
      A0
      D9 EXX      ;RESTORE [DE] AND SAVE OUR PLACE
      E0 RET PO   ;RETURN IF FINISHED
      D9 EXX      ;SAVE [DE] AGAIN AND RESTORE OUR PLACE
      18 JR,LOOP  ;CONTINUE
      F0

```

NOTICE THAT THIS ROUTINE REPLACES ALL THE LOOPS IN "REPACK" EXCEPT THE SEARCH LOOP IN LINE 120 AND THE FILL IN LINE 30. EVEN THE MOVE ROUTINE IS NO LONGER NEEDED!! HOWEVER, DON'T THINK THAT THE LESSON IS WASTED! IF YOU START "HOMEBREWING" BASICARTS, THE "ON BOARD" MOVE SUBROUTINE IS THE CORNERSTONE OF YOUR EFFORTS AND ABSOLUTELY ESSENTIAL TO THE SETTING UP OF ANY BASICART!! REGARDLESS OF WHICH BASIC YOUR OBJECT PROGRAM USES!!

THE ABOVE ROUTINE USES TWO MASKS TO ISOLATE THE TEXT (WHICH OCCUPIES THE ODD BITS) AND THE PICTURE (WHICH RESIDES IN THE EVEN ONES), THEN PUTS THE TWO TOGETHER AND DEPOSITS IT IN THE 1ST 4K, READY TO BE TAPED! OF COURSE THE BASIC PART OF "REPACK" (EB VERSION) WILL CHANGE BECAUSE SO MUCH IS NO LONGER NEEDED, AND OTHER FUNCTIONS CAN NOW BE INCLUDED (SUCH AS THE 300 BAUD "DUMPING" PROGRAM SHOWN ABOVE)! IT WOULD THEN LOOK SOMETHING LIKE:

```

>CLEAR ;CF=L. [G0]

>10 DATA A,>300+1,!9000,!9E20,>310+1;CLEAR ;PRINT " AWAITING INPUT ";PRINT
" OF TEXT
>20 CALL A;LC=0;GET%(B);IF LCGOSUB 190;GOTO 20
>30 CLEAR ;PRINT " LOAD PICTURE NOW
>40 LC=0;GET%(!A000);A=-32584;IF LCGOSUB 190;GOTO 40
>50 CALL D;FOR F=0TO !9F60STEP 2;IF %(F)E=(F+!7000)/2
>60 NEXT F
>70 CLEAR ;PRINT " 1 SAVE FOR BB";PRINT " 2 SAVE FOR AB";D=KP-50;IF (D<-1)+
(D>0)GOTO 70
>80 PRINT " DO YOU WANT";PRINT " FILE SEARCH?";GOSUB 180;G=KP-50;IF (G<-1)+
(G>0)GOTO 70
>90 IF DGOTO 200
>100 CLEAR ;F=A;CF=SM.;IF GFOR D=72TO 89;TV=32;TV=D;INPUT "=",%(F);F=F+2;
NEXT D;PRINT " IS THIS OK?";GOSUB 180;IF KP-49GOTO 100
>110 CF=L.;GOSUB 160;IF KPIF G PUT H,18
>120 PUT%(B),E;PRINT " PLAY TAPE BACK IN";PRINT " TO VERIFY";LC=0;VERIFY;
IF G VERIFY
>130 IF LCGOSUB 190;FOR F=0TO 2000;NEXT F;GOTO 110
>140 CLEAR ;PRINT ;PRINT " TO RERUN UTILITY";PRINT " PRESS [G0]";IF KP=13RUN
>150 STOP
>160 CLEAR ;PRINT " START TAPE RECORDING";PRINT " AND HIT ANY KEY
>170 RETURN
>180 PRINT " 1=YES 2=NO";RETURN
>190 PRINT " BAD LOAD TRY AGAIN";RETURN
>200 CLEAR ;IF GPRINT " INPUT FILE SEARCH CODE";A=KP-48;IF (A<1)+(A>9)GOTO 200
>210 IF GPRINT " IS THIS OK?";GOSUB 180;IF KP-49GOTO 200
>220 CLEAR ;GOSUB 160;NT=1;;PRINT ;IF KPPRINT ;IF GPRINT "IF G#",#1,A,"GOTO 70"
;TV=0;TV=A;TV=13
>230 FOR F=0TO 2000;NEXT F
>240 NT=6;E=20078;FOR F=C+4TO C+52STEP 2;E=E+2;PRINT "%(",#5,E,")=",%(F);NEXT F
>250 PRINT "&(9) (AND ITS COUSINS)=(WHATEVER)
>260 (OTHER LOOPS FOR MACHINE CODE AND ETC.)
>270 FOR F=1TO 999;NEXT F;CLEAR
>280 NT=1;PRINT "CLEAR ;FOR A=16384TO 19999;%(A)=KP;NEXT A;%(20050)=",#6,%(C),
";A=",#1,%(C+2),";:RETURN ;RUN
>290 NT=0;FOR F=BTO C-1;CY=0;TV=%(F);NEXT F;;RETURN ;CLEAR ;PRINT " DONE!";
STOP
>300 .FILL ROUTINE
>310 .OUR HOMEBUILT REPACK ROUTINE

```

NOTE: FOR BRB THE COMMAND "GET" BECOMES ":INPUT ", "PUT" BECOMES ":PRINT ", "VERIFY" BECOMES ":LIST ", THE "%(!A000);A=-32584" IN LINE 40 GETS CHANGED TO "%(!8000);A=27832", AND THE ":PRINT " IN LINE 220 BECOMES ":PRINT 300".

IN EITHER EB YOU'LL HAVE TO KEY IN LINES 300 AND 310 AS SHOWN, THEN "POKE" IN THE MACHINE CODE WITH:

```

>DA.A,>300+1,>310+1;FOR C=A TO A+9;INPUT %(C);NEXT C;FOR C=B TO B+25;INPUT %(C);
NEXT C

```

YOU CAN KEY IN THE HEX VALUES DIRECTLY IN EB BY PLACING AN EXCLAMATION POINT "!" BEFORE EACH NUMBER!

NOW, CAN YOU WRITE AN AB PROGRAM THAT USES OUR "HOMEMADE" MACHINE CODE "REPACK" ROUTINE? **IT CAN BE DONE!** HINT: DON'T FORGET AB'S ADDRESSING SCHEME!

SO, DID I GO ,TOO, FAST? IF THERE'S A QUESTION, CALL OR WRITE, MY ADDRESS AND ETC. IS IN THE "FILE SEARCH II" TUTORIAL (LAST TIME). NEXT TIME WE'LL LOOK AT &(9) AND ITS "COUSINS", CC=0, "BANK SWITCHING", THE "ON BOARD" MENU, AND BASICARTS! SO DON'T "PACK IT IN" YET!!