<table>
<thead>
<tr>
<th>NAME</th>
<th>TYPE</th>
<th>START</th>
<th>END</th>
<th>USED</th>
<th>FREE</th>
</tr>
</thead>
<tbody>
<tr>
<td>@DCSIFICF</td>
<td>P</td>
<td>00000008</td>
<td>0000008</td>
<td>00000000</td>
<td>00000000</td>
</tr>
<tr>
<td>@SYSMVBP</td>
<td>P</td>
<td>00000091</td>
<td>0000091</td>
<td>00000006</td>
<td>00000000</td>
</tr>
<tr>
<td>START</td>
<td>P</td>
<td>00000099</td>
<td>0000101</td>
<td>00000003</td>
<td>00000000</td>
</tr>
<tr>
<td>@MOVE1</td>
<td>P</td>
<td>00000102</td>
<td>0000141</td>
<td>00000039</td>
<td>00000001</td>
</tr>
<tr>
<td>@BACKUP</td>
<td>P</td>
<td>00000142</td>
<td>0000147</td>
<td>00000004</td>
<td>00000002</td>
</tr>
<tr>
<td>@TO.DISK</td>
<td>P</td>
<td>00000148</td>
<td>0000208</td>
<td>0000061</td>
<td>00000000</td>
</tr>
<tr>
<td>@RAMDISK</td>
<td>P</td>
<td>00000209</td>
<td>0000223</td>
<td>0000015</td>
<td>00000000</td>
</tr>
<tr>
<td>@DSAPPLY</td>
<td>P</td>
<td>00000224</td>
<td>0000230</td>
<td>0000007</td>
<td>00000000</td>
</tr>
<tr>
<td>@MENU</td>
<td>P</td>
<td>00000231</td>
<td>0000269</td>
<td>0000036</td>
<td>00000003</td>
</tr>
<tr>
<td>@FORMAT</td>
<td>P</td>
<td>00000270</td>
<td>0000334</td>
<td>0000065</td>
<td>00000000</td>
</tr>
<tr>
<td>@RESTOR</td>
<td>P</td>
<td>00000335</td>
<td>0000340</td>
<td>0000004</td>
<td>00000002</td>
</tr>
<tr>
<td>@DISK</td>
<td>P</td>
<td>00000341</td>
<td>0000348</td>
<td>0000005</td>
<td>00000003</td>
</tr>
<tr>
<td>@TO.CREO</td>
<td>P</td>
<td>00000349</td>
<td>0000354</td>
<td>0000006</td>
<td>00000000</td>
</tr>
<tr>
<td>@BACKUP</td>
<td>P</td>
<td>00000355</td>
<td>0000409</td>
<td>0000055</td>
<td>00000000</td>
</tr>
<tr>
<td>@VERCPU</td>
<td>P</td>
<td>00000410</td>
<td>0000412</td>
<td>0000003</td>
<td>00000000</td>
</tr>
<tr>
<td>@TOIMAGE</td>
<td>P</td>
<td>00000413</td>
<td>0000438</td>
<td>0000024</td>
<td>00000002</td>
</tr>
<tr>
<td>@DSFIGP</td>
<td>P</td>
<td>00000439</td>
<td>0000442</td>
<td>0000003</td>
<td>00000001</td>
</tr>
<tr>
<td>@DSFIG</td>
<td>P</td>
<td>00000443</td>
<td>0000555</td>
<td>0000110</td>
<td>00000003</td>
</tr>
<tr>
<td>@HITRATE</td>
<td>P</td>
<td>00000554</td>
<td>0000573</td>
<td>0000017</td>
<td>00000001</td>
</tr>
<tr>
<td>.STARTD</td>
<td>D</td>
<td>00000574</td>
<td>0000576</td>
<td>0000003</td>
<td>00000000</td>
</tr>
<tr>
<td>@MOVEFIL</td>
<td>P</td>
<td>00000577</td>
<td>0000645</td>
<td>0000067</td>
<td>00000002</td>
</tr>
<tr>
<td>@TO.CREF</td>
<td>P</td>
<td>00000646</td>
<td>0000703</td>
<td>0000056</td>
<td>00000002</td>
</tr>
<tr>
<td>@STAPEB</td>
<td>P</td>
<td>00000704</td>
<td>0000775</td>
<td>0000054</td>
<td>00000018</td>
</tr>
<tr>
<td>@STAPER</td>
<td>P</td>
<td>00000776</td>
<td>0000828</td>
<td>0000043</td>
<td>00000010</td>
</tr>
<tr>
<td>@RECOVER</td>
<td>P</td>
<td>00000829</td>
<td>0000882</td>
<td>0000054</td>
<td>00000000</td>
</tr>
<tr>
<td>@DSTAPEB</td>
<td>P</td>
<td>00000883</td>
<td>0000916</td>
<td>0000084</td>
<td>00000000</td>
</tr>
<tr>
<td>@VERCPUR</td>
<td>P</td>
<td>00000967</td>
<td>0000969</td>
<td>0000003</td>
<td>00000000</td>
</tr>
<tr>
<td>@VERCPUB</td>
<td>P</td>
<td>00000970</td>
<td>0000972</td>
<td>0000003</td>
<td>00000000</td>
</tr>
<tr>
<td>@DSTAPER</td>
<td>P</td>
<td>00000973</td>
<td>0001032</td>
<td>0000058</td>
<td>00000002</td>
</tr>
<tr>
<td>@TO.SUBS</td>
<td>P</td>
<td>00001033</td>
<td>0001073</td>
<td>0000041</td>
<td>00000000</td>
</tr>
</tbody>
</table>

R 114 used as of 2/2/94
DS/SCSI UTILITIES

START
CMENU
/about
STARTD
CSSNAVBP
CFORMAT
CMOVEFIL
CBACKUP
CVERCPUB
CSTAPEB
CDSTAPEB
CRESTORE
CVERCPUR
CSTAPER
CDSTAPER
CRECOVER
CTO.CREF
TOD.CREF
TCM.IMAGE
CTO.SUBS
CTO.DISK

DISK MANAGEMENT UTILITIES
1/20/94 Disk V1.1 1994

MAIN MENU

MOVING SELECTED FILES FROM A DISK INDEX

LVP BACKUP TO FLOPPY

220 APPEND NOT SUPPORTED. / ERASE TAPE (Y OR N):

APPEND ONTO TAPE OR ERASE TAPE (A or E)

BACKUP UTIL

RESTORE UTIL

INITIAL LOAD PROG

MAIN PART OF CREATING LIST
MOVING SELECTED FILES FROM A DISK INDEX

DISK MANAGEMENT UTIL MENU

11/15/92 CHANGED "SURFACES" TO "ADDRESSES" FOR PROTECT/UNPROTECT
11/23/92 REMOVED PERIODS FROM DS/MODD 7 IN QUESTIONS
11/25/92 CHANGED REFERENCES TO "SURFACES" IN ALL SCREENS TO "ADDRESSES".
7/27/93

DISK APPLY

CPSYSNAVBP
CFORMAT
CMOVEFIL
CBACKUP
CVERCPUB
CSTAPEB
CDSTAPEB
CRESTORE
CVERCPUR
CSTAPER
CDSTAPER
CRECOVER
CTO.CREF
TOD.CREF
TCM.IMAGE
CTO.SUBS
CTO.DISK

CRAMDISK
CVERCPUR
CSCS1CFG
CDSSAPPLY

EDIT DISK UTILITIES.
Utility Changes in Chronological Order

11/4/92 CGENPAR - UNABLE TO EXECUTE IF CURRENT CONFIG > 8.M.
11/5/92. CDSCFG - CHANGED ALL REFERENCES TO 'SURFACE TO ADDRESS' FOR PROTECT/UNPROTECT.
11/7/92 CPSTAT - ADDED MESSAGE ON SCREEN, 'SF 12/NEXT + SF 13/REV' FOR SUBSEQUENT SCREENS.
11/9/92 CSICSICFG - CHANGED SECTOR PROMPT MESSAGE 2. ADDED TEST TO DETERMINE IF SECTOR SIZE ENTERED EXCEEDS AVAILABLE SECTORS. CORRECTED ERROR MESSAGE FOR 13. 3. ADDED LINE 1280 TO CLEAR ERROR MESSAGE. 4. RESET SECTORS AVAILABLE TO FULL AVAILABILITY IF ENTER 'NO TO CONFIG.' ACCEPTABLE. 5. CHANGE LINE 1230 TO READ '# OF ADDRESSES TO ASSIGN'.
11/23/92 CDSCFG - REMOVED 'A & 'ADDED 'B TO LINPUTS ON LINES 4030,4132,4160,4215,8515
12/2/92 CDSCFG - WHEN RESPONaed 'NO TO APPLY CHANGES CURSOR IS RETURNED. CORRECTED TO SEND BACK TO THE BASE ADDR. SCREEN. ADDED WARNING, LINE 8150 & 8155, WHEN APPLYING CHANGE.
12/3/92 CDSCFG - ADDED WARNING FOR MAKE BACKUP WHEN KEY SF '10 TO APPLY CONFIG. CHANGED MESSAGE FOR APPLY PROCEEDS COMPLETED, LINE 9010.
12/4/92 CSICSICFG - ADDED 'FN/TAB, Exit TO ALL SCREENS,1020,1145,1225,1440. ADDED WARNING TO APPLY SCREEN, 1440.
12/11/92 CDSCFG - CHANGED ALL REFERENCES TO 'SURFACE TO ADDRESS'.
12/15/92 CDSCFG - CLEANED UP SCREENS FOR PASSWORD, CONFIG FILENAME, & ASSIGNMENTS; DS DEFAULTS DID NOT WORK, CORRECTED.
12/17/92 CSICSICFG - CHANGED ALL SECTORS FACTS FROM ADDRESS IF RESPONDED 'NO' TO CONFIGURATION ACCEPTABLE?'. MADE ENTRIES < 100 UNACCEPTABLE FOR SECTOR SIZE.
12/22/92 CSICSICFG - REMOVED 'Y' DEFAULT FROM ALL ENTRIES ACCEPTABLE?'. EXTENDED MESSAGE FOR CONFIGURATION FILENAME.
12/32 CTO.CREPO - CHANGED SCREEN HEADIG
15/93 CDSCFG - ADDED NOTE TO INDICATE IQS LEVEL NEEDED IF > 65535 SECTORS. ADDED CODE TO PREVENT OVERWRITE OF CONFIG FILE WITHOUT NOTICE.
12/21/93 CMOVEFIL - SPLIT LINE 290 TO ALLOW TRANSPARENT CONVERSION TO 'NEW' FORMAT.
2/1/93 CFORMAT - ADDED, LINE 617, TO INSURE EITHER DOS OR 2200 FORMAT SELECTED IF FLOPPY. CHANGED LINE 780 TO BRING YOU TO THE DISK ADDR SCREEN IF DISK SCRATCHED. 3- RESPONDED. NO TO CONTINUE.

2/8/93 CFORMAT - CORRECTED PROGRAM TO SHOW ERROR IMMEDIATELY IF FORMAT FAILED.

2/8/93 CFORMAT - DOS FORMAT WOULD NOT COME BACK, 'FORMAT COMPLETED: CORRECTED.' LINE 813, DISK UTILITIES 1.0

2/16/93 CSICSICF6 - CONFIGURATION SPENT WRONG ON LINE 160. A 350, CORRECTED.

3/3/93 CGENPART - CORRECTED 'INVALID CONFIGURATION FOR THIS CRU.' VLSI = 386.

3/9/93 CSICSICF6 - CAN'T FINISH CONFIGURING IF RUN OUT OF SECTORS. ENTERING 8 NOW. CAN BE USED TO END SECTOR ASSIGNMENT. ADDRESS SPENT WRONG ON LINE 610.

3/10/93 CSICSICF6 - IF BING CONFIG & RESTART SETUP, PASSWORD SHOWS UP IN SECTOR FIELD.

3/11/93 CSICSICF6 - IF MORE THAN 15 ADDR. ASSIGNED/1ST SLAVE ADDR IS LAST > ALL SUBSEQUENT ADDRESSES MOVE UP 1.

3/11/93 CSICSICF6 - INT. X71 ON LINE 660. CONVERTS TO '#(#)############

3/12/93 CFORMAT - IF DOS FORMAT CHOSEN WOULD NOT CHECK DISK FOR INDEX. DELETED LINE 705. LONG MESSAGES WOULD NOT COMPLETELY CLEAR. CHANGED LINE 7010.

3/16/93 CFORMAT - CHANGED VERSION TO 1.07.00.

3/16/93 CSCTAPER - FN/TAB TO EXIT DISAPPEARS IF NON-SCSI ADDR. NOT A TAPE ADDR.

3/16/93 CSCTAPERB - SAME AS ABOVE FOR GSCTAPER.

3/16/93 CSCTAPER - FN/TAB DISAPPEARS FROM SCREEN. WHEN LOADING DIRECTLY.

3/25/93 CDSCF6G - IF ENTERED # > 65535, SCREEN WOULD RUMP UP A LINE. ALL FOLLOWING ENTRIES WOULD BE ON WRONG LINE.


4/1/93 CMOVEFIL - ADDED TEST FOR 3 BYTE ADDRESS. NOT SUPPORTED.

4/1/93 CTO.CREF - ADDED TEST FOR 3 BYTE ADDRESS. NOT SUPPORTED.

4/1/93 CINSTALL - ADDED TEST FOR 3 BYTE ADDRESS. NOT SUPPORTED.

4/26/93 CFORMAT - IMPLEMENTED 3 BYTE ADDRESSING.
CFORMAT - made changes for V1.51 to ignore 3 byte changes.

CFORMAT - fix bug which caused 3 byte SCRATCH & corrected display on 3 byte index.

CFORMAT - fix bugs to allow formatting of Rx & LVP removable disks. Made correction to check address B10 for floppy type. Moved mount message to prevent overwrite of removable disk type.

CREF - changed verify on line 55 from (0,0) to (0,1) to circumvent Turbo bug.

CDSAPPLY - file is missing from DISK UTILITIES DISK REV 1.0, P/N 731-80158. Add to DISK UTILITIES 1.1 & to my master 1.0.

CDSCF1G - after keying SF'10 to begin procedure to apply a new configuration, after responding 3 instead of entering a remark for hard copy printout, program would next prematurely ask 'Apply Y or N', change word 'Apply' to 'Continue'.

CFORMAT - LVP DSPD comes up as Px type, corrected.

CDSCF1G - continue message overwrite table, fixed. Would not accept 'M' & 'S' for 'Master/Slave', corrected.

CDSCF1G - changed min OS requirements on line 4315 for 3 byte addressing.

CFORMAT - made ver 2.00 compatible to 386. Must now remove REM% on 935 for 3 byte scratch.

CDSFMVPB - changed heading to DISK UTILITIES 1.1 "...1994".

C.DISK - changed heading to "Disk Management Utilities. (Release 1.10)".

C.TYPE  - added message 'Device Not Supported' for non-Turbo.

CDSCF1G - corrected error message for format failure on drive B.

ACLOC - split line 470 creating 475 to allow conversion to 'NEW' format.

CDAVFU - split line 930 creating line 935 to allow conversion to 'NEW' format.

CDSCF1G - corrected DS defaults to use Master & Slave addresses like switches would.

Correct INPUT WARNING to react according to keys. Inserted line to recognize 32 MB MicroPac.

CDSCF1G - updated for OS/386 OS 1.3 with 3 diskettes & 70 files. Updated message to be more descriptive if OS/Disk not installed.
3/10/94  GINSTALL - UPDATED FOR CS/TURBO INSTALL W/ REL 1.30.01 FROM 1.2M OR 3 360K.

3/2/95  GINSTALL - ON TURBO WOULD INDICATE C386 AS O/S FILENAME ON SOURCE DISK
         WHEN SHOULD SAY CMVP.

3/2/95  CPSTAT - FAILS ON NON-386 CPUS W/ ERROR S19 ON LINE 28. CORRECTED.

4/2/95  CGENPART - COULD NOT ROOT MVP W/ 64K MEM. NO SF'15. IF LOADED
         CONFIGURATION W/ > 16 PARTITIONS ON CS MACHINE WOULD HANG IN LOOP. CORRECTED.

8/8/95  CDOSFORM - REMOVED \ AT END OF MESSAGE TO OPERATOR TO "AND
         STRIKE ENTER WHEN READY"

8/10/95  CDOS - ADDED ADDITIONAL SYNTAX CHECKING FOR FORMAT COMMAND TO VERIFY
          EITHER ‘DRIVE A OR B SELECTED AND ’ IS PRESENT.

8/11/95  CGENPART - ADDED LINE TO ALLOW BYPASS OF ‘RECONFIGURATION PASSWORD’
          ON AUTO EXECUTION.

11/4/95  CCLOC - FIXED TO SHOW CORRECT YEARS FROM 1971 TO 2090

11/4/95  CCLOCK - FIXED TO SHOW CORRECT YEARS FROM 1991 TO 2090

2/20/98  CSYSMPB - ADDED DISPLAY CLOCK TO MENU

5/8/99  CBCBACKUP - UPDATED TO ACCEPT YEAR 00
3/8/99 Updated to accept year $\emptyset$. B

DELETED: IF $U_9 = \emptyset$ THEN 920 from end of line 900
1/28/94 Would not convert to NEW format. Line 470 too long.
   Fix: Split line 470 creating line 475

11/14/97 Only display years 1900 through 1999 on calendar.
   Fix: 550 IF M <> INT(M) OR M < 10 OR M > 12 THEN 530: IF D <> INT(D) OR D < 10
         OR > 31 THEN 530: IF Y <> INT(Y) OR Y < 0 THEN 530: IF Y < 100 THEN Y = Y + 1900
   Change to: 550 :: IF Y > 99 THEN Y = Y + 1900; IF Y < 91 THEN Y = Y + 2000

11/14/97 Only display years 1900 through 1999. Hangs in loop at transition from 1999 to 2000.
   Fix: DELETE line 110
         110 IF D$ <> "000101" THEN 115: GOTO 100
   4050 IF M <> INT(M) OR M < 1 OR M > 12 THEN 4030: IF D <> INT(D) OR D < 1
         OR D > 31 THEN 4030: IF Y <> INT(Y) OR Y < 0 THEN 4030:
         IF Y < 100 THEN Y = Y + 1900
   Change to: 4050 :: IF Y > 99 THEN Y = Y + 1900; IF Y < 91 THEN Y = Y + 2000
NOTES

11B: IF D8<>"000101" THEN 115: $BREAK: GOTO 120 100

DELETE LINE 11B

4050: :: IF Y < 100 THEN Y = Y + 1900
CHANGE TO 4050: :: IF Y > 90 THEN Y = Y + 1900: IF Y < 91 THEN Y = Y + 2000

JANUARY 1, 2000 is a SATURDAY

145

JAN 1 20*** SAT
FEB 1  TUE

G:   CLOC
550: :: IF Y = 90 THEN Y = Y + 1900: IF Y < 91 THEN Y = Y + 2000

✓ CHECKED THESE YRS TO INSURE DAY OF MONTH CORRECT

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Tu</td>
<td>Tu</td>
<td>M</td>
<td>Sr</td>
<td>Su</td>
<td>F</td>
</tr>
<tr>
<td>2</td>
<td>F</td>
<td>W</td>
<td>T</td>
<td></td>
<td>Su</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>W</td>
<td>W</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Su</td>
<td>Su</td>
<td>Sr</td>
<td>Su</td>
<td>Th</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Tu</td>
<td>Tu</td>
<td>Su</td>
<td>M</td>
<td>F</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>W</td>
<td>W</td>
<td>T</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Th</td>
<td>Th</td>
<td>T</td>
<td></td>
<td>Su</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>F</td>
<td>F</td>
<td>Th</td>
<td>T</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Su</td>
<td>F</td>
<td>Th</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>Tu</td>
<td>M</td>
<td>Sr</td>
<td>F</td>
<td>Th</td>
<td></td>
</tr>
</tbody>
</table>
LIST "DATE" 20 30 50 80 160 230 240 290 370 430 450

20&U7$(.) - ENTERED DATE TIME MMDDYYHHMMSS
REM U9$() - DISPLAYED DATE TIME MMDDYYHHMMSS
30 REM U8$(1) - ENTER TIME U8$(2) ENTER DATE MM/DD/YY V3$ DATE FROM DATEFILE
50
80 V3$ = DATE
160
230 DATE = V3$ PASSWORD U6$
240
290
370 V2$ = DATE
430 V3$ = DATE

550 IF Y < 100 THEN Y = Y + 1900

560 IF Y > 90 THEN Y = Y + 1900:
      IF Y < 90 THEN Y = Y + 2000

12/31/99 - 1/1/2000

TIME HANGS ON CLOCK PROG

2/28/00 - 2/29
2/29/00 - 3/1
12/31/00 - 1/1

CLOCK 4050 IF Y < 100 THEN Y = Y + 1900:
         IF Y > 90 THEN Y = Y + 1900:
         IF Y < 90 THEN Y = Y + 2000

12/31/99 - 1/1/2000
2/28/00 - 2/29/2000
12/31/2001 - 1/1/2001

TIME HANGS AT 00:00:00

12/31/99 - 1/1/2090
12/31/99 - 1/1/2091

YEAR GOES BACK TO 1991
Interoffice Memo

Date: 05/15/97
To: Jerry Crawford
CC: George Reinhart
From: Mike Bahia
Subject: Year 2000 and 2200 Systems

Jerry,

Completed testing on the 3 different 2200 Operating Systems for the year 2000. None of the 3 Operating Systems, Basic-2 O/S 3.5 for the standard 2200, CS/386 Release 1.30.00, or Turbo Release 1.30.01 use the date or time under normal operating conditions. Both date & time are system variables which can be used by the programmer in any way they would see fit. As with many systems, the date variable is 6 characters in length in the format of YYMMDD. Any program making decisions based on the last 2 digits of the year could present a problem.

The "Initialize Date & Time" Utility program (@CLOC) which comes with all 3 operating systems was used to implement the transition testing. With each system type, a Clock Program (@CLOCK) which displays the time & date & includes the current month & following month, was run on a 2nd terminal. Various programs were executing simultaneously on other terminals including a diagnostic program to test the system instruction set, a system benchmark test, a CRT exerciser, 2200 Word Processing, and a system game, Martians. Only the 2 clock related programs showed any ill affects. On all 3 Operating Systems, the following date transitions were tested with results as shown:

<table>
<thead>
<tr>
<th>Transition</th>
<th>Inialize Date/Time (@CLOC)</th>
<th>@CLOCK</th>
</tr>
</thead>
<tbody>
<tr>
<td>12/31/99 to 01/01/00</td>
<td>No problem</td>
<td>Clock stops at 00:00:00, Program goes into endless loop</td>
</tr>
<tr>
<td>02/28/00 to 02/29/00</td>
<td>Year shown is 1900</td>
<td>Year shown is 1900</td>
</tr>
<tr>
<td>02/29/00 to 03/01/00</td>
<td>Year shown is 1900</td>
<td>Year shown is 1900</td>
</tr>
<tr>
<td>12/31/00 to 01/01/01</td>
<td>Years shown are 1900/1901</td>
<td>Years show are 1900/1901</td>
</tr>
</tbody>
</table>

Both programs will be fixed to correctly show the year as will the clock problem with @CLOCK stopping at 00:00:00 on transition to year 2000. The fixes should be fairly simple. If you have any questions or need further information please let me know.

05/15/97
Confidential
Leap Years

The Rule

According to the Gregorian calendar, which is the civil calendar in use today, years evenly divisible by 4 are leap years, with the exception of centurial years that are not evenly divisible by 400. Therefore, the years 1700, 1800, 1900 and 2100 are not leap years, but 1600, 2000, and 2400 are leap years.

Background

The Gregorian calendar year is intended to be of the same length as the cycle of the seasons. However, the cycle of the seasons, technically known as the tropical year, is approximately 365.2422 days. Since a calendar year consists of an integral number of whole days, a calendar year cannot exactly match the tropical year. If the calendar year always consisted of 365 days, it would be short of the tropical year by about 0.2422 days every year. Over a century, the calendar and the seasons would depart by about 24 days, so that the beginning of spring in the northern hemisphere would shift from March 20 to April 13.

To synchronize the calendar and tropical years, leap days are periodically added to the calendar, forming leap years. If a leap day is added every fourth year, the average length of the calendar year is 365.25 days. This was the basis of the Julian calendar, introduced by Julius Caesar in 46 B.C. In this case the calendar year is longer than the tropical year by about 0.0078 days. Over a century this difference accumulates to a little over three quarters of a day. From the time of Julius Caesar to the sixteenth century A.D., the beginning of spring shifted from March 23 to March 11. When Pope Gregory XIII instituted the Gregorian calendar in 1582, the calendar was shifted to make the beginning of spring fall on March 21 and a new system of leap days was introduced. Instead of intercalating a leap day every fourth year, 97 leap days would be introduced every 400 years, according to the rule given above. Thus, the average Gregorian calendar year is 365.2425 days in length. This agrees to within a half a minute of the length of the tropical year. It will take about 3300 years before the Gregorian calendar is as much as one day out of step with the seasons.
THE GREGORIAN CALENDAR AND LEAP YEARS

The current year, 1996, is referred to as a "leap year" because we have inserted a "leap day" to make the length 366 days rather than the usual 365 days. The official name of the "leap day" is an intercalary day (with the accent on the second syllable). Intercalary is the adjective form of the verb to intercalate, which means to insert. Once this day is inserted, or intercalated, it becomes an embolistic day. Still another name for this extra day every fourth year is the bissextile day, meaning a double sixth day.

This last name is derived from the location of the intercalary day every fourth year in the Julian calendar that was put into use by Julius Caesar in 45 B.C. He realized that a tropical year, the interval of time between successive beginnings of spring, is about 365 1/4 days. Since agriculture was the main occupation of most people in the world until 200 years ago, adjusting the calendar year to fit the seasons seems the most reasonable way to form a calendar. The best one can do is to make the average length of a year 365 1/4 days by having 366 days in every fourth year. Caesar's intercalary day was inserted on what is now the day before February 24, thus making a second, sixth day (bisextile) counting back from March 1 -- the beginning of the Roman year. (The Roman days of the month were not numbered as in our modern calendar.) In 533 A.D., when the monk Dionysius Exiguus determined the year of Christ's birth and began the numbering system for years A.D., it worked out conveniently that years evenly divisible by four were leap years.

Unfortunately, 365 1/4 days is not the exact length of a tropical year. The time between successive instants of the beginning of spring is 365 days, 5 hours, 48 minutes, and 46.02 seconds. While this makes the Julian year only 11 minutes and 13.92 seconds longer than a tropical year, it amounts to spring beginning one day earlier after about 128 years. The seasons were slowly moving backward through the year under the Julian calendar.

\[1582,\text{ Pope Gregory XIII was informed by his astronomer, Christopher Clavius, that the first day of spring had fallen on March 10 of that year. Since the first day of spring had fallen on March 21 in 325 A.D. when the Council of Nicaea had established the dates for the holidays in the Christian calendar, Gregory felt it was important to put the seasons back in the same places in the calendar. He therefore declared that the next day after October 4, 1582, was October 15, 1582. He also adopted a revision in the calendar which resulted in the average length of a year being closer to the length of a tropical year. This calendar, which we now use, is called the Gregorian calendar. It differs from the Julian calendar by letting century years, such as 1600, 1700, 1800, etc., be leap years only if they are evenly divisible by 400. Thus, 1600 was a leap year, and 2000 will be, too, but 1700, 1800, and 1900 were not leap years.}

The average length of our Gregorian year is 365 days, 5 hours, 49 minutes, and 12 seconds, still about 26 seconds longer than a tropical year. However, more than 3000 years must now pass for the seasons to move back by one day. A suggested modification to the Gregorian calendar is to eliminate as leap years those years which are evenly divisible by 4000. This would result in the length of a year, averaged over 4000 years, being only 4 seconds longer than a tropical year. Fortunately, we still have a couple thousand years before we need to give serious thought to this modification.
1/28/94 WOULD NOT CONVERT TO NEW FORMAT. LINE 930 TOO LONG.

FIX: SPLIT LINE 930 CREATING LINE 935.
DOS COPY

FAILS W/ DISK ON LINE 180 OF
C: \ DOSCOPY TRY TO COPY A
FILE FROM A FLOPPY TO A DOS DISK

COPY A: C:MAXEO B: C:MAXEO

WORKS IF COPY FROM E:0
COPY C: E:MAXEO B: E:MAXEO

USGS C: DOSCOPY
**IMPORTANT MESSAGE**

TO ____________________________

DATE ____________________________ TIME ____________________________ A.M. P.M.

WHILE YOU WERE OUT

M ____________________________

OF ____________________________

Area Code & Exchange ____________________________

<table>
<thead>
<tr>
<th>TELEPHONED</th>
<th>PLEASE CALL</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALLED TO SEE YOU</td>
<td>WILL CALL AGAIN</td>
</tr>
<tr>
<td>WANTS TO SEE YOU</td>
<td>URGENT</td>
</tr>
<tr>
<td>RETURNED YOUR CALL</td>
<td></td>
</tr>
</tbody>
</table>

Message

__________________________
__________________________
__________________________
__________________________

Operator ____________________________
CDOS

8/10/95

If type in wrong syntax or illegal drive parameter with FORMAT command, does not give an error & defaults to the A drive.

Fix: Added additional syntax checking for FORMAT command on line 460 with additional error messages.

460 IF STR(B$8,2)="" THEN 896: IF STR(B$8,1)<>"A" AND STR(B$8,1)<>"B" THEN 895: IF STR(B$8,2)<>"A:" AND STR(B$8,2)<>"B:" THEN 891: LOAD T "CDOSFORM"

895 PRINT HEX(0D 0E 07); "INVALID DRIVE SPECIFICATION (ONLY DRIVE A OR B SUPPORTED WITH FORMAT)"; HEX(0F): GOTO 180

896 PRINT HEX(0D 0E 07); "REQUIRED PARAMETER MISSING (DRIVE DESIGNATION & COLON)"; HEX(0F): GOTO 180

897 PRINT HEX(0D 0E 07); "PARAMETER FORMAT NOT CORRECT (DRIVE DESIGNATION & COLON)"; HEX(0F): GOTO 180
1/27/94 If fails formatting Drive B could return message "General Failure Error reading Drive A".
Fix: Changed line 480 as follows:

480 % General Read Error or bad Write

8/8/95 If try to format B, comes back with message to "Insert new diskette for drive A; and Strike ENTER when ready". Would always default to drive A or B if drive letter incorrect. Added syntax checking for Format command & removed \ at end of message.
470 % and strike ENTER when ready
Package Subject: DOS Utilities

Item Title: DOS Utilities

Mike
I will check the error when using drive B. There is a hidden key to bring the last command back. This is done by pressing Cancel/Editor key then $F'3$ on 2200 keyboard or sf'4 on a PC emulation which is SF'3 in reality.

Regards John Baxi

--------------------------------------------- Original Memo ---------------------------------------------

To: Kirit Baxi  From: Mike Bahia
Subject: DOS Utilities  Date Sent: 10/08/92

John,

During the last few days I have been doing some testing with the DOS Utilities. While it was fresh on my mind I had a couple of suggestions.

1. In testing FORMAT, I was using drive B. Was formatting a 360K disk in a 1.2 Meg which I knew was illegal. However, it did go out to format the disk but when it failed it indicated a failure with drive A. The message was, "General Failure error reading drive A" "Abort, Retry, Fail?"

I believe this should have indicated the error was reading drive B. Drive A is address D30 and has the DOS Utilities as found on the latest Turbo O/S, 1.18. Drive B is D20. Also, what does Fail mean in the 'Abort, Retry, Fail?' line. It seems to be the same as Abort. Doesn't seem to be necessary.

2. Sometimes when you do get an error you need to rerun the Utility to get the prompt back. It would be nice to have a SF key to bring the prompt back. What do you think?

PS: Initially had a lot of problems because I did not realize a space was not needed between the drive designation and the filename for those commands that use both. This was causing most of my problems. This is where a SF key to bring back the DOS prompt would have been handy.

Best regards,

Mike
Line/Command Trap Ver for Marine Biological Labs

CDSTAPEB 1/3/92  Rel 2.6

LIST "1430" 810,840,850,870,880,920,940,960,1000,1020,1030,1130,1160,1200,1240,1250,1330

810 REM % REWIND TAPE: GOSUB '201 ("ReWinding TApe"): $G10REWIND TAPE#1: AB = "REWIND": L = 810: IF STR$(g$,6,3) <> $HEX$(00 00 00) THEN 1430

840 REM % ERASE TAPE: GOSUB '201 ("Erasing TApe"): $G10ERASE TAPE#1: AB = "ERASE": L = 840: IF STR$(g$,6,3) <> $HEX$(00 00 00) THEN 1430

850 REM % POSITION TO BEGINNING OF TAPE DIRECTORY: GOSUB '201 ("Positioning to Tape Directory"): STR$(g$,3,3) = $HEX$(00 00 00): $G10SEEK TAPE DIRECTORY BLOCK#1: AB = "SEEK DIR": L = 850: IF STR$(g$,6,3) <> $HEX$(00 00 00) THEN 1430

870 REM WRITE BLOCK TO TAPE BUFFER: GOSUB '203: AB = "WR DIR LABEL": L = 870: IF STR$(g$,6,3) <> $HEX$(00 00 00) THEN 1430

880 REM WRITE FILE MARK: $G10WRITE FILE MARK#1: AB = "WR DIR FM": L = 880: IF STR$(g$,6,3) <> $HEX$(00 00 00) THEN 1430

920 REM % RETENSION TAPE: GOSUB '50 (HEX$(5E) "$RTENSIONING TApe"): $G10RETENSION TAPE#1: AB = "RTENSION": L = 920: IF STR$(g$,6,3) <> $HEX$(00 00 00) THEN 1430: TB = "T"

940 REM % POSITION TO BEGINNING OF TAPE DIRECTORY: GOSUB '201 ("Positioning to Tape Directory"): STR$(g$,3,3) = $HEX$(00 00 00): $G10SEEK TAPE DIRECTORY BLOCK#1: AB = "SEEK DIR": L = 940: IF STR$(g$,6,3) <> $HEX$(00 00 00) THEN 1430

160 REM READ TAPE DIRECTORY LABEL (1st Block): GOSUB '202: AB = "READ DIR LABEL": L = 960: IF STR$(g$,6,3) <> $HEX$(00 00 00) THEN 1430

1000 REM CHECK FOR END OF DATA: AB = "END OF DATA": L = 1000: IF STR$(g$,6,3) <> $HEX$(17 00 00) THEN 1430

020 M$ = "POSITIONING TO LAST BLOCK": GOSUB '201(M$): STR$(g$,3,3) = CA: IF CA <> $HEX$(00 00 00) THEN STR$(g$,3,3) = SUBCHEX$(00 00 01): IF CA < $HEX$(00 3C 00) THEN STR$(g$,3,3) = $HEX$(00 00 00): $G10SEEK BLOCK#1: AB = "SEEK LAST BLK": L = 1020: IF STR$(g$,6,3) <> $HEX$(00 00 00) THEN 1430
1030 L\# = "POSITION TO END OF DATA"; GOSUB '201(M\#); G10. SEEK END OF DATA#1:
   \# = "SEEK END DATA"; L = 1030: IF STR(08, 3) <> HEX(00 00 00) THEN 1430
1130 G10. BACKUP SECTORS#1: \# = "BACKUP SECTORS"; L = 1130: IF STR(08, 4) <> HEX(00 00 00) THEN 1430
1166 REM WRITE BLOCK TO TAPE BUFFER: GOSUB '203: \# = "WR BLK"; L = 1166: IF STR(08, 3) <> HEX(00 00 00) THEN 1430: NEXT 2
1200 REM WRITE BLOCK TO TAPE BUFFER: GOSUB '203: \# = "WR BLK"; L = 1200: IF STR(08, 3) <> HEX(00 00 00) THEN 1430
1240 REM WRITE BLOCK TO TAPE BUFFER: GOSUB '203: \# = "WR BLK"; L = 1240: IF STR(08, 3) <> HEX(00 00 00) THEN 1430
1250 REM % WRITE FILE MARK: G10. WRITE FILE MARK#1: \# = "WR DATA FM"; L = 1250: IF STR(08, 3) <> HEX(00 00 00) THEN 1430
1330 REM % REWIND TAPE: GOSUB '201("Rewinding Tape"); G10. REWIND TAPE#1: \# = "REWIND TAPE";
   L = 1330: IF STR(08, 3) <> HEX(00 00 00) THEN 1430

1430 IF STR(08, 7, 2) = HEX(00 00) THEN 1490: \# = "TAPE COMMAND ERROR"; GOTO 1700
1700 REM % NONRECOVERABLE ERROR: \# = "TAPE BACKUP ABORTED"; IF 03% <> "" THEN \# = M18""; CORP /"\% D3%"; GOSUB '50 (\#; \#; \#; PRINT HEX(07); PRINT AT (20, 0); "LAST TAPE FUNCTION"; \#; " LINE " J; " ERROR CODE = " ; STR(08, 1); " COMMAND ERROR = " ; STR(08, 7, 2)" END: 
1705 PRINT AT (21, 0); " STR(08, 3) = " : HEXPRINT STR(08, 6, 3): PRINT AT (21, 25); " ERROR CODE IN HEX = " : HEXPRINT STR(08, 6, 1): PRINT AT (21, 50); " COMMAND ERROR IN HEX = " : HEXPRINT STR(08, 7, 2)
11/15/92 Changed all references to 'Surface' in all screens to 'Address'.
Lines 90, 350, 1145, 4030, 4290, & 8515.

11/23/92 Removed periods from DS and added '?' to questions.
Lines 4030, 4132, 4160, 4215, 8515.

12/2/92 If enter 'N' when asking if you want to 'Apply Y or N?'
To Appyling Configuration program returns ':. Same response from any key but 'Y'.
Changed line 8080 as follows:

Was 8080 K$="" : PRINTAT(21,0,80) ; LINPUT "Apply Y or N" =KB ; IF KB ="Y" THEN RETURN
To 8080 K$="" ; LINPUT "Apply Y or N" =KB ; IF KB ="Y" THEN 8150 ;
IF KB ="N" THEN 8080 : GOTO 40

Added warning when applying changes. Added line 8150 & 8155.

12/3/92 Added warning to backup when key SF 10 to Apply Config.
Split line 8010, making 8012. Put message on 8010.
Changed message for Apply Procedure completed. See line 9010.

12/5/92 Moved password to print at line 2 instead of 1. See line 8020.
Moved 'Config filename' to print at line 2 instead of 1. See line 8030.
Added space after 'Assignments' to remove the word Exit.
Using default config gave illegal message. Changed IF/THEN

Address from 4384 to 4390 on line 4132 to match DS Util 2.04 which Works.
1/15/93  Added note to indicate if > 65535 sectors assigned to disk may 0/S 386 1.2 or Turbo 1.1 required. Created new line 4315 IF X < 65536 THEN 4320: PRINT HEX(01,0E); AT(23,0); "MIN 0/S 386 1.2
or Turbo 1.1 required w/ > 65535." Added PRINT AT(23,0,55)
At beginning of line 4290 to clear 0/S level message.

Added code to prevent overwriting of old config files without notice.
New lines 4452 IF E <= 2 THEN 4440: KA = "": PRINT AT(23,0,55); " Data file "; FB; " already exists."; LINPUT " Overwrite, Y/N?" -KA
4453 IF KA = "N" OR KA = "Y" THEN 4440: IF KA = "Y" OR KA = "Y" THEN 4455:
GOTO 4452

Due to PRINTAT problem with Turbo which confuses screen if sector value exceeds 65535 added Turbo test exit on line 4315

After first:

JH = BPSSTAT(1): IF STR(JH,9,1) = "T" THEN 4320:

3/25/93  IF entered a sector value > 65535 on a 386 or VLSI, screen would
Bump up 1 line & all future sector entries would not line up with proper address.
Also value > 65535 not accepted on VLSI or 386, 0/S now. Most recent,
4315 IF X: JH: IF STR: PRINT HEX(01,0E); AT(23,0); "MIN 0/S 386 1.2 Turbo 1.1
For > 65535. " ; ; GOTO 4295

3/26/93  Deleted line 4155 which put unneeded message: "Run - Accept Screen?" on CRT when asking for master or slave address.

7/27/93  Change line 8080 substituting Continue for the word Apply in the LINPUT
statement. Apply was being prematurely used, after message for hard 0/S.
In process of apply procedures, after responding to 'Hard copy to printer' and remark, the screen updates with 'Continue Y or N?' but overwrites the last line of addresses. Due to. Change line:

```
8080 K$ = " " : PRINT AT(23,0,80) ; : INPUT ..............
```

Master/Slave won't accept small s/m.

Change line:

```
4145 A$ = " " : IF KB = " M " OR K$ = " m " THEN A$ = " Master " : IF K$ = " S " OR K$ = " S " THEN A$ = " Slave " : IF A$ = " " THEN 4160 : PRINT AT(20,13) ; A$ ; ADD 4167 IF A$ = " Master " THEN K$ = " m " : IF A$ = " Slave " THEN K$ = " S "
```

Changed line 4315 to indicate minimum D/Sfel for 3 byte addressing is Turbo 1.386x or 1.25:

```
4315 IF X < 65536 THEN 4320 : J$ = AP$STAT(1) : IF STR(J$,9,1) = " T " THEN 4320 : PRINT HEX(010E) ; AT(23,0) ; MM D/S Turbo 1.30.01 or 1.25 for > 65535. " j : GOTO 4295
```

If using a 140M drive w/ "Use DS Defaults", w/ Z" drive both could be configured on the master side. 4 addresses would be lost. Set up defaults so drives 3-4 always will come up as slave addresses, I-Z as master address, same as older formats using switches. Also do not verify existence of drive select 1.

Fix: 4390 REM: : : FOR I = 1 TO 4 : IF WZ#(I) = 1 THEN 4398 : IF WZ#(I) = 1 THEN

```
4394 IF I = 1 THEN STR(WZ#(I),6) = HEX(01) : IF I = 2 THEN DO : STR(WZ#(I),6) = STR(WZ#(I),4) ADDHEX(01) : ENDDO
```

```
4391 IF I = 3 THEN STR(WZ#(I),6) = HEX(01) : IF I = 4 THEN DO : STR(WZ#(I),6) = HEX(01) : IF STR(WZ#(3),6) = HEX(01) THEN STR(WZ#(I),6) = STR(WZ#(3),4) ADDHEX(01) : ENDDO : KB = STR(WZ#(I),6) : IF KB = " " THEN KB = KB ADDHEX(C6) : X = MAX(VAL(KB),1) : REM
```
4394 NEXT I: GOTO 4400
4398 STOP "ILLEGAL CONFIGURATION!! MUST HAVE A DRIVE 1 TO CONFIGURE.
WINCHESTER DRIVES NEED TO BE PROPERLY CONFIGURED. CONTACT CONFIGURATION BEFORE PRECEEDING."

AT WARNING WHEN STOPPING THROUGH APPLY ROUTING, ANYTHING BUT A CAPITAL 'Y'
RESPONSE TO CONTINUE RETURNS YOU TO BASE ADDRESS SCREEN.
Fix 8150 DIM I#:1: PRINT "...: INPUT "CONTINUE, Y OR N?" - I#: IF I# = "Y" OR I# = "Y" THEN 8160
8155 IF I# = "N" OR I# = "n" THEN 40: GOTO 8150

32M MICROPOLIS MAY NOT BE RECOGNIZED AS A CORRECT SW SETTING FOR SOME
PORT VERSIONS. PUT IN FOLLOWING LINE WITH A REM AS DID NOT HAVE A 32M
MICROPOLIS TO TEST.

1116 REM IF P=9 THEN P=6: REST OF LINE UNCHANGED.
CDSCFG16 BUGS

3/25/93 If use DS Defaults twice in a row using the same configuration, file name fails 2nd time w/ message "Illegal Sector Address or No Platelet". Happens on Turbo, 386, & VLSI.

Circumvention: Open & shut floppy disk or reset and rerun.

3/25/93 If use existing configuration file & respond NO to overwrite screen bumps up a line. (Turbo, 386, & VLSI)

3/26/93 Master/Slave won't accept small s/m. Fixed line 416S or if kib = 512MB

3/25/93 With configured 140M on Drive Select 1 & unconfigured 64M on Drive Select 3, when used DS Defaults assigned addresses D11-D14 to the 64M (drive 3) & addresses D15-D1E to the 140M.

If make 64M drive 1 & 140M drive 3, drive 3 is assigned addresses D11-D1E & the 64M is lost (using DS Defaults). Fix line 4390, 4391 2/2/94

9/16/93 When preparing to 'apply' a created configuration after responding to 'Hard copy to printer?' & 'Remake for hard copy?' 'Continue Y or N?" Wipes out last line of addresses. (D & E line)

Fix: Change line 8080

9/16/93 Use 64M as Drive 1 & 140 M as drive 3 when making copies for 64M & several surfaces for 140. All master addresses from D25-D2D. Only applied address to drive 3. When looking at config 64M appears as unconfigured & 140M as address D21-D29 instead of D25-D29.

NPP IF change addresses of existing drives good idea to power off first to clear cache which can cause weird problems. Fixed 2/2/94
2. No 32 M not recognized ADD LINE IF P = 9 THEN P = 6 AT 1116

1170 SETS UP WZA & WJ ADDRESSES FOR DRIVES 1 START ADDR
CASEDEFT - BOTH 64 & 140

HOOKS CORRECT
D21 16M
D22 16M
D23 16M
D24 16M
D61 10M
D2E 10M

APPLIED BOTH \rightarrow SEEMS OK
APPLIED DRIVE 3 ONLY \rightarrow SEEMS OK
APPLIED DRIVE 1 ONLY \rightarrow SEEMS OK

CASE 2 64M D21-D24 ALL 16M 140M D25-D2E 16M D2D 16M

LOOKS CORRECT
D61 I91 SEEMS CORRECT

140M NOW DRIVE 1  64M NOW DRIVE 3

CONFIG SHOWS 140M  D21-D28 16M  D29 6M  64M  D2A-D2D 16M

LOOKS OK w/ CDSCFG6  LISTS SEEM CORRECT
DS DEFAULTS 140M 64
D21 10M  D2B 16M
D2A 10M  D2E 16M

CONF & SCREENS MATCH  LIST MATCHES CONF

64M SHOULD BE D61-D64
140M SHOULD BE D21-D2E
CREATE DRIVE CONFIG: BYTES 001 - 032 CONSTANT: 01 - 013 = "TBO": 14 # OF PLATTERS: 15 START ADDR

BYTES 033 - 256 14 BYTE ENTRY | DR# | 565 + #MMN: WHERE #(N, 3, 3): #MMN = # OF SECTORS

W5#32 OR SURFACE DRIVE 1, 2, 3, 4: W#(32) 3 BY SURFACE # OF SECTORS

PRINT

F# = "CDEFAULT": LINPUT "ALGNAM": F# CASE 6

LIMITS T#T, F#, A, B, C, E: E = ERR

IF E < 2 THEN 4455

IF E = 1: 4440

IF E = 0: DATASAVE DC OPEN T#T, 6 F# CASE 6 OPEN DATAFILE CASE 6

LIMITS T#T, F#, A, B, C, E:

DATALOAD DC OPEN T#T, F#

W1 = A

FOR W = 1 TO 4

KB = 8IN W

INIT (00) R%(): X = 0

FOR I = 1 TO 31

IF ST# (W5#, I, 1) = K# THEN DO

X = X + 1

NEXT I

WRITE SECTOR SIZE FOR EA ADDR INTO R#
25 DIM WZ%(4)
190 WZ$ = ""   
326 WZ$(WZ) = BIN(WZ,3) & BIN(P+1) & G11 & K
1170 WZ$(I) = BIN(Z,3) & BIN(P+1) & STR(D1$(I),1) & STR(D1$(I),5,1)
4381 STR (WZ$(I),6,1) = K
4390 IF WZ$(I) = " " THEN 4394
4392 W$(X+Z-1) = STR(WZ$(I),3)
USE DS DEFAULTS?

HALT STEP

64 M = DRIVE 1

140 M = DRIVE 3

4322 IF K# = "Y" THEN 4390

4390 REM SET W$(32) = \[1\] sizes

INIT (00) W$(1) , W$A : MAT WO: ZER

FOR I = 1 TO 4

IF WZ$(I) = "" THEN 4394 NO DRIVE

K$ = STR (WZ$(I), 6)

START ADDR 0! FOR DRIVE 1

IF K$ > " " THEN K$ = K$ ADD HEX (CE)

X = MAX (VAL (K$), 1)

X = 1 FIRST ADDR

4392 FOR Z = 1 TO VAL (STR (WZ$(I), 4)) # OF ADDRS FOR CURRENT DRIVE

W$(X + Z - 1) = STR (WZ$(I), 3)

W$(X + Z - 1) = W$(X + Z - 1) + 6502.4

IF STR (DI$(I), 1, 1) <= "Z" THEN STR (W$A, X + Z - 1, 1) = BIN (I)

NEXT Z

W$(I) = VAL (STR (WZ$(I), 5)) - 47

W$(I) = 7

4391 NEXT I

4390 I = 2

IF WZ$(I) = "" THEN 4394 NO DRIVE 2

4394 NEXT

4390 I = 3

IF WZ$(I) = "" THEN 4394

K$ = STR (WZ$(I), 6)

START ADDR 05 FOR DRIVE 3

IF K$ > " " THEN K$ = K$ ADD HEX (CE)

K$ = 05

X = MAX (VAL (K$), 1)

X = 5

FOR Z = 1 TO VAL (STR (WZ$(I), 4))

W$(X + Z - 1) = STR (WZ$(I), 3)

W$(X + Z - 1) = W$(X + Z - 1) + 6502.4

IF STR (DI$(I), 1, 1) <= "Z" THEN STR (W$A, X + Z - 1, 1) = BIN 1
ADDDRO 1746 IF P = 9 THEN P = 6

SPLIT 4890
4391 IF

A WARNING CONTINUE THEN ANYTHING BUT Y TAKE YOU TO BASE AODE

8150
6155

SWAPPED 140 TO DRIVE 1 (HAD ADDR D61-D66 & 64M TO DRIVE 3 (HAD ADDR D21-D28)

HAD CABLES WRONG ONCE & SIN SETTINGS

RUN PRG

SET SF 15 TO START SETUP

USE DEFAULTS Y

FILE NAME CASE & RET

8432 : : : W$(J) = STR$(R$(Z+1),4)

J = 33
Proposed DS Address Assignments

2 Winchester with sectors available

<table>
<thead>
<tr>
<th>Master disk Address</th>
<th>Catalog Maximum</th>
<th>Slave disk Address</th>
<th>Catalog Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>D21 1</td>
<td>45024</td>
<td>D31 3</td>
<td>38912</td>
</tr>
<tr>
<td>D22 1</td>
<td>45024</td>
<td>D62 3</td>
<td>38912</td>
</tr>
<tr>
<td>D23 1</td>
<td>45024</td>
<td>D63 3</td>
<td>38912</td>
</tr>
<tr>
<td>D24 1</td>
<td>45024</td>
<td>D64 3</td>
<td>38912</td>
</tr>
<tr>
<td>D25 0</td>
<td>0</td>
<td>D65 3</td>
<td>38912</td>
</tr>
<tr>
<td>D26 0</td>
<td>0</td>
<td>D66 3</td>
<td>38912</td>
</tr>
<tr>
<td>D27 0</td>
<td>0</td>
<td>D67 3</td>
<td>38912</td>
</tr>
<tr>
<td>D28 0</td>
<td>0</td>
<td>D68 3</td>
<td>38912</td>
</tr>
<tr>
<td>D29 0</td>
<td>0</td>
<td>D67 3</td>
<td>38912</td>
</tr>
<tr>
<td>D30 0</td>
<td>0</td>
<td>D67 3</td>
<td>38912</td>
</tr>
<tr>
<td>D31 0</td>
<td>0</td>
<td>D67 3</td>
<td>38912</td>
</tr>
<tr>
<td>D32 0</td>
<td>0</td>
<td>D67 3</td>
<td>38912</td>
</tr>
<tr>
<td>D33 0</td>
<td>0</td>
<td>D67 3</td>
<td>38912</td>
</tr>
<tr>
<td>D34 0</td>
<td>0</td>
<td>D67 3</td>
<td>38912</td>
</tr>
<tr>
<td>D35 0</td>
<td>0</td>
<td>D67 3</td>
<td>38912</td>
</tr>
</tbody>
</table>

Configuration created in file CASEDEFT
Key Reset and shift '10 to apply FN/TAB - Exit
Proposed DS Address Assignments
2 Winchester drives available

Data as read from file: Write this for drive 1

```
01000000000000000000000000000000
2F000000000000000000000000000000
000000FED0000000000000000000000
000000FED0000000000000000000000
000000FED0000000000000000000000
000000FED0000000000000000000000
000000FED0000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
```

Try Y or N.
Proposed 8 & Address Assignments
2 Winchester with sectors available

Data as read from file

write this for drive 3

Apply Y or N Y
Proposed DS Address Assignments

2 Winchester disks available

<table>
<thead>
<tr>
<th>Master disk Address</th>
<th>Catalog Maximum</th>
<th>Slave disk Address</th>
<th>Catalog Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>D21 1</td>
<td>65024</td>
<td>D61 0</td>
<td>0</td>
</tr>
<tr>
<td>D22 1</td>
<td>65024</td>
<td>D62 0</td>
<td>0</td>
</tr>
<tr>
<td>D23 1</td>
<td>65024</td>
<td>D63 0</td>
<td>0</td>
</tr>
<tr>
<td>D24 1</td>
<td>65024</td>
<td>D64 0</td>
<td>0</td>
</tr>
<tr>
<td>D25 3</td>
<td>65024</td>
<td>D65 C</td>
<td>0</td>
</tr>
<tr>
<td>D26 3</td>
<td>65024</td>
<td>D66 0</td>
<td>0</td>
</tr>
<tr>
<td>D27 3</td>
<td>45024</td>
<td>D67 0</td>
<td>0</td>
</tr>
<tr>
<td>D28 3</td>
<td>45024</td>
<td>D68 0</td>
<td>0</td>
</tr>
<tr>
<td>D29 3</td>
<td>45024</td>
<td>D69 0</td>
<td>0</td>
</tr>
<tr>
<td>D2A 3</td>
<td>65024</td>
<td>D5A 0</td>
<td>0</td>
</tr>
<tr>
<td>D2B 3</td>
<td>65024</td>
<td>D5B 0</td>
<td>0</td>
</tr>
<tr>
<td>D2C 3</td>
<td>65024</td>
<td>D5C 0</td>
<td>0</td>
</tr>
<tr>
<td>D2D 3</td>
<td>24576</td>
<td>D6D 0</td>
<td>0</td>
</tr>
<tr>
<td>D2E 0</td>
<td>0</td>
<td>D6E 0</td>
<td>0</td>
</tr>
</tbody>
</table>
D21-4
64M

Master disk  Catalog
Address      Maximum

Slave disk  Catalog
Address      Maximum

Write this for drive 1

SAME AS CASE DEFT

Apply Y or N.
<table>
<thead>
<tr>
<th>Master disk</th>
<th>Catalog Address</th>
<th>Maximum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slave disk</td>
<td>Catalog Address</td>
<td>Maximum</td>
</tr>
</tbody>
</table>

Data as read from file:

```
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
00000000000000000000000000000000
```

Write this for drive 3

Apply Y or N Y
STOP 190  DRIVE 1 = CM  DRIVE 3 = 140 M

RUN
BASE ADDR D20  RETURN
GETS DEVICE STATUS
STOP 190  W2H = ALL(20)
INIT(00) W7H()
W2A() = ""  ALL 20
MAT W = ZER . W2. W3 = 0
200 IF R(QH) = "01" THEN V1 = 1  R0H = 40  PROM VER 1.0
IF V0 = 1 THEN GOSUB 3010  V = φ
210 IF R(QH) <= "32" THEN 250  PROM VER > 1.0 ≤ 32
GOSUB 1020  GET DRIVE SW STATUS
1020 G0 = STR(PH, 1, 1) OR HEX(20)
1030 EB, STR(GH, 2, 7) = ALL(00)

#GIDREADDRIVESTATUS#2-1  (GH)G1; STR(EH, VAL(STR(GH, 5, 1)))

1040 PRINT
1050  D@H() = E@H
  FOR I = 2 TO 5
  D1B(I-1) = D@H(I) & BIN(I-1) & HEX(0000)
  NEXT I
1060 REM MOVE WINC INTO D1B()
1070 REM
1080 REM
1085 NL, N2 = 0
  FOR I = 1 TO 4
  D@H = D1B(I)  ZL  ZL  ZL  ZL
  X, Z = VAL(STR(D@H, 3))  XH  Z  2  2
  IF Z = φ THEN 1100
  IF Z > 4 THEN Z = 2. 64
  IF STR(D@H, 1) < "  THEN 1090 X = " 6"  IS IT A WING DRIVE 3-10?
1090 IF X > 15 THEN 1095  X = 4  FOR SLAVE ADDR (> 15)?
START ADDR

1100 GOTO 1100

1110 FOR I = 1 TO 4

1120 IF (I * 8) = 1 THEN STOP ERROR

1130 IF (I * 8) = 9 THEN DRIVE UNCONFIGURED

1140 IF (I * 8) = 9 THEN NO DRIVE

1150 IF (I * 8) = 9 THEN BAD DRIVE

1160 IF (I * 8) = 9 THEN ILLEGAL SET

1170 IF (I * 8) = 9 THEN SET

1180 IF (I * 8) = 9 THEN CONVERT

1190 NEXT I

1200 IF (I * 8) = 9 THEN Z = 38912:

1210 IF (I * 8) = 9 THEN Z = 65024

1220 IF (I * 8) = 9 THEN Z = 20 DIM D$(16)40

1230 IF (I * 8) = 9 THEN ERROR Z = Y

1240 IF (I * 8) = 9 THEN FOR I = 1 TO 4

1250 IF (I * 8) = 9 THEN NEXT I

1260 IF (I * 8) = 9 THEN FOR I = 1 TO 4

1270 IF (I * 8) = 9 THEN NEXT I

1280 IF (I * 8) = 9 THEN FOR I = 1 TO 4

1290 IF (I * 8) = 9 THEN NEXT I

1300 IF (I * 8) = 9 THEN RETURN

210 REM

250 REM DISPLAY INFO ON DRIVES
APPLYING CASES

LOAD "CDSAPPLY" 9020,9999 BEX 9005

0500 INIT(00) W3B = 000000

GOSUB 8250

8250 INIT(00) RB()

DATALOADBAT #1, (A,A) RB() : READ CONFIG DATA

2520 X = 0

IF STR(RA(1),11,3) <> "TBD" THEN X = 1: = TBD

IF RA(3) = HEX(00-00) THEN X = 1:

9005 IF X = 0 THEN GOSUB 9020

9020 REM

9025 BIN(STR(RA(1),11,3)) = W : W = 1 FOR DRIVE

9030 PRINT READING:

FOR X = 1 - 16

PRINT AT (3+X,1); HEXOF(RA(X)) PRINT 1ST 16 BYTES PRINTS

NEXT X

9040 I = VAL(STR(RA(1),16)) I = 8 LAST BYTE 1ST

IF I > 0 THEN STR(RA(1),16,4) = CH(I)

CS(i) = 0EFF0806 REM APPLY 4 CONST BYTES

9050 FOR X = 3 TO 16 PRINT sectors for each addr

IF STR(RA(X),4,3) > HEX(00 00 00) THEN STR(RA(X),4,3) = W3B

ADD C STR(W3B/3), STR(RA(X),4,3)

NEXT X ADDR FIELDS

9060 K# = K# ADD STR(RA(1),X) NEXT 256 TIMES

AT END HEXPRINT KA = 210 0001

DO YOU WANT TO APPLY

9005 NEXT W W = Z

INIT(00) W3B: GOSUB 8250

8250 INIT(00) RB()

DATALOADBAT #1, (A,A) RA() ALL DD

8252 X = 0

IF STR(RA(1),11,3) <> "TBD" THEN X = 1

IF RA(3) = HEX(00-00) THEN X = 1

RETURN

9005 IF X = 0 THEN 9020

NEXT W 3
## DEFAULTS

CASE 3 1/28/94

Proposed DS Address Assignments
2 Winchesters with sectors available

<table>
<thead>
<tr>
<th>Master Disk</th>
<th>Catalog</th>
<th>Slave disk</th>
<th>Catalog</th>
</tr>
</thead>
<tbody>
<tr>
<td>Address</td>
<td>Maximum</td>
<td>Address</td>
<td>Maximum</td>
</tr>
<tr>
<td>D21</td>
<td>1</td>
<td>D61</td>
<td>0</td>
</tr>
<tr>
<td>D22</td>
<td>1</td>
<td>D62</td>
<td>0</td>
</tr>
<tr>
<td>D23</td>
<td>1</td>
<td>D63</td>
<td>0</td>
</tr>
<tr>
<td>D24</td>
<td>1</td>
<td>D64</td>
<td>0</td>
</tr>
<tr>
<td>D25</td>
<td>1</td>
<td>D65</td>
<td>0</td>
</tr>
<tr>
<td>D26</td>
<td>1</td>
<td>D66</td>
<td>0</td>
</tr>
<tr>
<td>D27</td>
<td>1</td>
<td>D67</td>
<td>0</td>
</tr>
<tr>
<td>D28</td>
<td>1</td>
<td>D68</td>
<td>0</td>
</tr>
<tr>
<td>D29</td>
<td>1</td>
<td>D69</td>
<td>0</td>
</tr>
</tbody>
</table>

CASE 5 CREATED

- DRIVE 1
  - 140M
  - 45024
- DRIVE 2
  - 65024
  - 9526
  - D24
  - D29
  - 65024
  - D25
DISK STATUS
CS-D R3 PROM L2M DIO
2275 360K DZO
DS RH PROM L2M DZO
2275 360K D10 (VLSI)
PHOENIX D10 (VLSI)
LVP FLOPPY D10

G6()
2/1/93 added line 617 to insure either DOS or 2200 format was selected if using 5 1/4" floppy.

NEW LINE 617 IF A2# = "2" THEN 620: IF A2# = "1" THEN 620: GOTO 615

If formatting an address which had already been formatted and you answer no to whether or not to continue format program would take you back to mount platter screen. Change program to take you back to the address screen if responded with 'N'.

Change line 780

780:...: IF KA = "N" THEN 510:

was IF KA = "N" THEN 610:

2/5/93 when format failed, error did not display, went back to either DOS or 2200 format selection if floppy or mount disk. Then after key return error shows in lower left corner.

Made line 9040 to 9050 error GOSUB return line

Copied line 7010 to 9040 prints error

Split line 610 creating line 612

On 610 added 610 $CLOSE #1: IF M<0 THEN 612:

On 612 added 612 PRINT AT (9,0,80): M2 = 1: A2# = "1": IF A1# = "D10" OR A1# = "D20"

or A1# = "D30" THEN 615: GOTO 620

2/8/93 when using DOS format with floppy would not come back - say "Format Computed". Came back to main menu.

Changed line 813 to GOTO L 820 if successful, not line 815.

813 IF STR(G$, 6, 3) = HEX(00 00 00) THEN (815) 820
3/12/93 CF\textsc{format} does not recognize if the diskette has data on it when a 
\textsc{dos\ format\ is\ chosen.}

\textbf{Fix:} delete line 705 \textsc{if a2$\textasciitilde$ = "2" then 795}

\textsc{sometimes long messages are not cleared \at the bottom of screen.}

\textbf{Fix:} 7010 \textsc{print at}(23,0,33); \textsc{hex}(oe); \textsc{str(ma)}; \textsc{hex}(01); \textsc{rem \- display message}

\textsc{7010 print at}(23,0,50);}

\textbf{line 110. changed version to 1.07.00 \from 1.06.00.}
OLD FORMAT 3.4

FORMAT COMPLETED, SCRATCH
NO MSG, WENT BACK TO ADDR

FORMAT COMPLETED
NO MSG, BACK TO ADDL

OK
NO MSG

OK
NO MSG

NEW FORMAT

VLSI 12275
CS 2200
DOS
WINCHESTER
3.86/DS 1.2 R3.8

FORMAT COMPLETED, SCRATCH
FORMAT COMPLETED

CS 2200
DOS
WIN
TURBO 12275
CS 2200
DOS

TURBO DS 1.2 3.5
CS 2200
DOS
SCSI

FORMAT COMPLETED
FORMAT COMPLETED
FORMAT COMPLETED
FORMAT COMPLETED

OK

OK

GOOD
GOOD
GOOD
785 REM \GET NEW INDEX PARAMETERS \ SCRATCH THE DISK
790 PRINT AT(16,60);"Enter new index information and press RUN": REM - CHANGE THE PROMPT
795 IF STR$(1)$="HEX(0)" THEN VS(2)="REM": ELSE VS(2)="OLD": REM - INITIALIZE THE STRUCTURE FIELD
800 PRINT AT(15,29);"Index Sectors = ":HEX(0); STR$(8)$;HEX(0);AT(15,29);"End Cat. Area = ":HEX(0);STR$(8)$;HEX(0);AT(17,29);"Structure = ":HEX(0);STR$(8)$;HEX(0);AT(21,29);"END OF CATALOGED AREA
805 PRINT AT(21,52);"RETURN - Next Field";AT(22,52);"RUN - Accept parameters": REM - ADDITIONAL KEY DEFINITIONS
810 END 0: GOTO 610
815 GOTO 510
820 GOSUB '201("Format Completed")": REM - WE MADE IT!
822 F=2
825 REM \GET NEW INDEX PARAMETERS \ SCRATCH THE DISK
830 PRINT AT(16,60);"Enter new index information and press RUN": REM - CHANGE THE PROMPT
835 IF STR$(1)$="HEX(0)" THEN VS(2)="REM": ELSE VS(2)="OLD": REM - INITIALIZE THE STRUCTURE FIELD
840 PRINT AT(15,29);"Index Sectors = ":HEX(0);STR$(8)$;HEX(0);AT(15,29);"End Cat. Area = ":HEX(0);STR$(8)$;HEX(0);AT(17,29);"Structure = ":HEX(0);STR$(8)$;HEX(0);AT(21,29);"END OF CATALOGED AREA
845 PRINT AT(21,52);"RETURN - Next Field";AT(22,52);"RUN - Accept parameters": REM - ADDITIONAL KEY DEFINITIONS
850 END 0: GOTO 610
855 REM - RETURN FILE COUNTER
860 REM \GOSUB '211": REM - SEARCH THE MESSAGE (IF ANY)
865 GOSUB '201("Help")": REM - PROCESS THE CURRENT FIELD
260 DIM V$(3) 8: REM
310 J$ = $PSTAT(1)

761 IF STR$(S$(1), 1, 1) <> HEX$(02) THEN 765: IF STR$(S$(2), 2, 2) = HEX$(0000) THEN 800: REM - IF 3 BYTE INDEX = 00000 THEN FORMAT IT
762 CONVERT VAL (STR$(S$(1), 2, 2)) TO V$(1), (######): CONVTR
763 PRINT AT (14, 29); INDEX Type = 3 Byte: GOTO 775

776 IF STR$(S$(1), 1, 1) = HEX$(01) THEN PRINT AT (15, 51); "": IF STR$(S$(1), 1, 1) = HEX$(02) THEN PRINT AT (15, 51); "8"

840 PRINT AT ..........: (Old or New) 880
841 IF STR$(S$(9, 1) <> "T" THEN 845 IF T <= 215 AND Z$ <= "3F" THEN 845:

885 GOSUB 'Z02: BTRAN(V$(2), 'Aa-Za 3" THEN RETURN: GOSUB 'Z01("Index Structure Must be 'OLD' or 'NEW' or 'TURBO Optionally '3')"

880 GOSUB 'Z02 (15, 145, V$(1), 8, "#", HEX$(7F 7F): Y$(1) = IF $(1): RETURN: REM
885 GOSUB 'Z02 (17, 45, V$(2), 3, "A", HEX$(7F 7E)): :: IF Y$(2) = "OLD" OR V$(2) = "NEW" OR V$(2) = "TRI": THEN RETURN
800 GOSUB 'Z01("Index Structure Must be 'OLD' or 'NEW' or 'TURBO Optionally 'TRI':"

821 GOSUB 300: REM - CHECK DISK TYPE

8300 DEFFN 300: REM %t, GET DISK STATUS

8320 DIM ZH1, XB1, XB1(1) ZR: REM - DISK TYPE VARIABLES (8 BYTES CHECK)

8020 BS=0:R8=ALL(20):STR(BS,1)=STR(BR,1) AND HEX(33):SELECT # 1< BS >

8030 G, G8=GET DISK TYPE # 2(020003OF12 2206000700 70A04000870B, G, (G))

8040 T=POS(HEX(C0D0)=TH):IF T<7 THEN RETURN

8050 REM %t, CHECK FOR DS CABINET

8060 G=HEX(30):STR(GS,7)=ALL(00):6108 STATUS REQUEST # 2 (0B, 400012 220600070070

A0400288D070406A06816H00087051A00C340, G8) G8; STR(XB1, VAL(STR(GS,5,1)))

8070 T=VAL(XB1)-48:IF T<4 THEN RETURN: REM - RETURN: IF NOT A DS

8080 G=STR(GS,7)=ALL(00):6108 READ

8090 IF STR(XB1, 7)=HEX(00) THEN STR(E8, 18, 1)=HEX(10):G1=STR(GS,8,1) AND HEX(10):IF G1>HEX(00) THEN 8120

8100 IF STR(GS,8,1)>HEX(00) THEN 8130:XB1(1)=XB1

8110 ZH1=STR(XB1, 4, 2): REM - ZH1 = DS PROM REVISION

8115 RETURN

8120 GOSUB 255("Disk Unavailable")

8125 RETURN

8130 GOSUB 255("Specified disk is not a DS")

8135 RETURN

87% "REVISED = 26 APRIL 1993 (MB) B: Drive Recognition Added"

110 V$8: "02.00.00" : REM - VERSION NUMBER

520 &C058 #1: PRINT TAB(20); (C) COPR. WANG LABS, INC. 1993: PRINT TAB(3); "REL " ; YB
885 GOSUB 201 (17, 45, VB(Z), 3, "A", HEX (7F 7E)): VB(Z) = IP$: (i) = $TRAN(VA(Z), "Aa-Zz")$R

886 IF VB(Z) = "OLD" OR VB(Z) = "NEW" THEN RETURN: IF VB(Z) = "TRI" AND STR(J$9, 1) = "T" THEN RETURN

887 IF STR(J$9, 1) = "T" THEN 888: GOSUB 201 ("INDEX STRUCTURE MUST BE 'OLD' OR 'NEW'"): PRINT HEX(07): GOTO 885: REM - ACCEPT INDEX STRUCTURE

888 GOSUB 201 ("INDEX STRUCTURE MUST BE 'OLD', 'NEW', OR 'TRI' FOR 3 BYTE"): PRINT HEX(07): GOTO 885: REM - ACCEPT INDEX STRUCTURE FOR TURBO

911 IF VB(Z) = "TRI" AND STR(J$9, 1) = "T" AND STR(J$9, 3) = "0" AND ZB = "3F" THEN 913: REM - 913 IF 3 BYTE TURBO, NON-0 ADD;

912 IF VB(Z) = "TRI" AND STR(J$9, 1) = "T" THEN 913: IF VI >= 1 AND VI <= 255 THEN 915: PRINT HEX(07): GOSUB 201 ("INDEX SECTORS MUST BE FROM 1 TO 255"): GOTO 850: REM - VALIDATE INDEX SECTORS VALUE

913 IF VI >= 1 AND VI <= 65535 THEN 915: PRINT HEX(07): GOSUB 201 ("INDEX SECTORS MUST BE FROM 1 TO 65535 FOR 3 BYTE"): GOTO 850: REM - VALIDATE INDEX SECTORS VALUE FOR 3 BYTE

917 IF VI <= V3 THEN 918: PRINT HEX(07): GOSUB 201 ("END CAT. AREA LESS THAN INDEX SECTORS"): GOTO 850

918 IF VB(Z) = "TRI" AND STR(J$9, 1) = "T" THEN 920: IF V3 = VI AND VB = "65535" THEN 920: PRINT HEX(07): GOSUB 201 ("END CAT. AREA MUST BE LESS THAN 65535"): GOTO 850: REM - VALIDATE 2 BYTE END CAT. AREA

933 IF STR(J$9, 3) = "0" THEN 940: IF T < -15 AND ZB < "3F" THEN 940:

935 IF VB(Z) <> "TRI" AND STR(J$9, 1) <> "T" THEN 940: SELECT 3 ON: SCRATCH DISK & T#1, LS = VI, END = V3: ERROR GOSUB 255: GOSUB 201 (M$): GOTO 850: REM - BUILD THE 3 BYTE INDEX STRUCTURE & HANDLE ERROR

934 IF < 256 THEN 935: GOSUB 201 ("BE PATIENT, LARGE INDICES TAKE TIME")
10 REM @FORMAT 03/20/90 -- (c) Copr. Wang Laboratories, Inc. 1986
20 REM (c) Copr. Wang Laboratories, Inc. 1986
30 REM
40 REM Program Name = @FORMAT
50 REM Author = Steve Mcllarry
60 REM Date Written = 22 April 1986
70 REM Last Revised = 20 March 1990 (TBD)
80 REM Last Revised = 10 February 1993 (MFB)
90 REM Last Revised = 12 March 1993 (MFB)
97 REM Revised = 26 April 1993 (MFB) 3-Byte Recognition added.
100 REM
105 DIME A1%, A2$, I, S1, D1, D2%, D3%
110 REM % VAREABLE DEFINITIONS
115 DIME V1, V2, V3: V1 = "00, 00, 00": REM - VERSION NUMBER
120 DIME D1 K, P1$: REM - DEVICE/PLATTER ADDRESS VARIABLES
125 DIME E1(40), E2(15): REM - STATUS/IO VARIABLES
130 DIME K$: REM - KEYIN BYTE
135 DIME M$: REM - MESSAGE ("201"
140 DIME M$: REM - ERROR HANDLER WORK VARIABLE
145 DIME I%, J%, T1$, T2$, T3$, T4$: REM - VARIABLES FOR ACCEPTING A FIELD
150 DIME I%, J%, T1$, T2$, T3$, T4$: REM - LOOP INDEXES
155 DIME M$: REM - GENERAL WORK VARIABLE
160 DIME M$, M$: REM - DISK MENU VARIABLES ("210"
165 DIME D1, D2$: REM - DISK SIZE, DESCRIPTION NUMBER ("222"
170 DIME D1, D2$: REM - "OTHER" DISK ADDRESS
175 DIME D1, D2$: REM - INDEX SECTOR BUFFER
180 DIME D1, D2$: REM - INDEX SECTOR DATA CONVERSION VARIABLES
185 DIME V1, V2$: REM - INDEX INFORMATION VALUES
190 DIME Z1(32), X1(32), Y1(32): REM - DISK TYPE CHECK VARIABLES FOR 3-BYTE
195 DIME G1(32), E1(32), F1(32), G1(32), E2(15), D1, D2$: REM - DISK STATUS
200 DIME D1, D2$: REM - DISK STATUS
205 REM % Initialize everything: REM % 1st display
310 J1$: ="STAT1"
320 SELECT PRINT /O05(80): PRINT HEX(02, 0D, 0C, 03, 0F)
330 $CLOSE #: PRINT HEX(03, 06, 0E); TAB(15): "SOFTWARE FORMATTABLE DISK PLATTER FORMAT UILITY": PRINT TAB(20); "(c) Copr. Wang Laboratories, Inc. 1993": PRINT TAB(31): "Release "; V1$: 
340 PRINT AT(22, 32); "RETURN - Proceed"; AT(23, 32); "FN/TAB - Previous Menu"; HEX(01)
350 REM % Get Disk Address
355 PRINT AT(4, 0): HEX(02, 04, 02, 04, 0F); "Enter platter address: ": HEX(0E); D0$: HEX(02, 04, 02, 04, 0F)
365 F$=0: PRINT AT(4, 22); LINPUT HEX(0E), -DO$:
370 REM % Show Platter information: REM Unhjav disk and show 2nd screen
375 REM % Show Platter information: REM Unhjav disk and show 2nd screen
385 REM % Show Platter information: REM Unhjav disk and show 2nd screen
390 GOSUB '201(" "): PRINT HEX(07): GOTO 370
400 REM % Show Platter information: REM Unhjav disk and show 2nd screen
410 $CLOSE #: PRINT HEX(03, 06, 0E); TAB(15): "SOFTWARE FORMATTABLE DISK PLATTER FORMAT UILITY": PRINT TAB(30); "Platter Information": PRINT AT(4, 0); "Platter address = "; DO$
420 PRINT AT(9, 0, 80): M2$="1": A2$="1": IF A1$="D1G" OR A1$="D20" OR A1$="D30" THEN 610
430 IF A2$="2": GOTO 620: IF A2$="1": GOTO 615
440 PRINT AT(9, 0); HEX(0E); "Mount disk to be formatted and press RETURN:"; REM -
450 REM - DISPLACE MESSAGE (IF ANY)
460 GOSUB '201(" "): REM - DISPLAY MESSAGE (IF ANY)
REM %1. Get disk status
7340 IF #8 = HEX(7E) OR #8 = HEX(2F) THEN K# = "R"
7350 IF K# = "Y" AND K# = "R" THEN...
7360 RETURN

8000 DEFFN '399: REM %1. Get disk status
8020 #8 = #8: A$ = ALL (G): I$ = STR$(#8, 2, 1): #8 = HEX(33): T$ = POS (#8, 0) = #8(9)
8050 IF #8 > #8(9) THEN RETURN
8060 REM %1. Check for DS cabinet
8100 #8 = HEX(#8): T$ = STR$(#8, 5, 1): #8 = HEX(09): IF #8 = 0 THEN G: #8 = #8(3, 1)
8110 IF #8 > #8(3, 1) THEN #8 = #8(18, 1) = HEX(09): G1# = STR$(#8, 8, 1) AND HEX(10)
8120 GOSUB '215("Specified disk is not a DS")
8130 RETURN
8150 GOSUB '215("Disk unavailable")
8160 RETURN

9000 DEFFN '255: REM % ERROR HANDLER (SETE M#=ERROR DESCRIPTION & KEPS)
9005 M# = ERR: REM - GET THE ERROR NUMBER
9010 M$ = "": REM - INITIALIZE MESSAGE
9015 IF M$ < " " THEN 9020: M$ = "": REM - GET MESSAGE IF DISK ERROR
9020 DATA "Disk Hardware Error","Disk Drive Not Ready","Disk Drive Time-Out","Disk Format Error","Disk Format Key Engaged": REM - DISK ERRORS
9021 DATA "Disk Seek Error or Platter Protected","Disk CRC Error","Disk LRC Error","Bad Sector Address/Platter Not Mounted","Verify Error": REM - DISK ERRORS (CON)
9025 IF M$ = "Illegal Device Specification" THEN 9040: M$ = "": REM - CATCH-ALL
9040 PRINT AT(23, 0, 33): HEX(0E) + STR$(M$): HEX(01): REM - DISPLAY MESSAGE
9050 PRINT HEX(07): RETURN
FORMAT ver. 2.00.00

827 GOSUB 300: REM

827 GOSUB 250: REM - CHECK DISK TYPE

8000 DEFFN 250: REM% ↑. GET DISK STATUS

LINE 935, SELECT 3 ON CAUSES 519 ERRORS ON VLSI+386.1.12
SCRATCHDISK & T#1, LS=V1, END=V3 CAUSES 516 ERROR ON VLSI+386.1.12

REMOVING THE REM% IN FRONT OF SELECT 3 ON + SCRATCHDISK IS OK

5/17/83 IF CREATE A 3 BYTE ADDRESS ON A TURBO THEN REFORMAT + SCRATCH WITH AN "OLD" OR "NEW" INDEX, REMAINS AS A 3 BYTE INDEX. REMOVING THE REM% IS OK.

FIX: CHANGE ↓

FROM: 933 IF STR(S#8,3)="0" THEN 940: IF T<>-15 AND Z#<3 THEN 940: IF VB(2)<>
   "TRI" AND STR(J#,9,1)<="T" THEN 940

TO 933

   "TRI" OR STR(J#,9,1)<="T" THEN 940

WHEN READING A 3 BYTE INDEX ON A DISK TO BE FORMATTED, THE "END CAT. AREA"
+ "CURRENT END" FIELDS ARE 1 LARGER THAN THEY SHOULD BE.

FIX: ADD THE FOLLOWING CHANGE:

762 CONVERT VAL (STR(S#1,2,2),2) TO VB (1), (####)
   CONVERT VAL (STR(S#1,4,3),3) -1 TO VB (2), (####)
   CONVERT VAL (STR(S#1,7,3),3) -1 TO VB (3), (####)
   REM - CHANGE THE 3-BYTE VALUES TO STRINGS
6/11/93  CANNOT FORMAT Phoenix removable, asks you to re-enter address.
Fix 615 ON T + 1 GOTO 565, 565, 620

- Added message as follows if tried to format Z270:
  5260 K1# = "Z270 type not supported" : RETURN

- Address B10 not recognized as floppy address and fails to check type.
  Fix 5030 IF POS ................ THEN 5035: IF STR (SA, 3) = "3" THEN STR (SA, 3) = "1": STR (SA, 1) = "D": A1# = SB

- Does not recognize LVP floppy & will not allow format.
  Re-enter address. Show message "Unknown Disk Type, check connector".
  Fix
  5090 TB = G8(11): T = POS (HEX (CO, DO) = TH): IF TH = HEX (20) THEN T = 5:
  IF T = 0 THEN 5140: IF T = 2 THEN GOSUB 5150.
  5100 ON T GOSUB 5260, 5250, 5240, 5230, 5270: PRINT ■ K1#
  5270 K1# = "LVP Floppy": RETURN
  615 GOSUB 5040: ON T + 1 GOTO 565, 565, 620: IF T = 5 THEN 620: A# = "360 KB";
  IF VAL (STR (EB, 4, 3)) = 4160 THEN A# = "1.2 MB"
  617 PRINT AT (5, 0, 80); HEX (OE); STR (A#, LEN (A#)); " Floppy Drive. Please Select:";
  LINOUT " A28: PRINT HEX (OE):

Mount message overwrites drive type (shown only w/ removable address)
Fix 620 PRINT AT (10, 0); HEX (OE); "Mount disk to be formatted and press RETURN:" ; REM - PROMPT

8/4/93  LVP DSDD comes up as Phoenix type.
5100 IF T = 2 AND STR (GB, 3) = HEX (10) THEN T = 5: ON T GOSUB 5260, 5250, 5240, 5230, 5270; PRINT KB
2/20/94 Error 519 on line 935 on CS|386. Does not recognize SELECT 3 on following REM 999. Error 516 on line 935 on CS|386. Does not recognize SCRATCHDISKS following 2 REM 999.

Fix: remove 2 nd % from both REM commands on line 935

Added REM on line 92 to remove REM 999's on line 935 for 3 byte scratch on Turbo.
CGENPART

1f current partition > 8. M. unable to execute w/ SF'15. Get
message "PARTITION TOO LARGE".

On Line 610 change: IF X > A(1) . . . TO . . . IF X > 59

3/2/93 Always get "INVALID CONFIGURATION" on VLSI system when booting from
a configuration for the 2nd time. Create always works. Problem caused because
byte 22 in the 1st sector of c:SYSFILE for current = 00 hex. Must be 20 hex
for VLSI.

Fixed: added line 11 DIM J#(1); J# = STR(C#9,1)

2760 GOSUB'191: IF 01<4 AND F0<2 THEN UB = "": IF J# = "M"
           THEN 2770

2765 M1# = HEX(00 40 80 00): FOR X = 1 TO 4: GOSUB'193(X): NEXT X;
       GOTO 2775

2770 M1# = HEX(20 40 80 00): FOR X = 1 TO 4: GOSUB'193(X): NEXT X
        2775 RETURN

3/3/93 If loaded turbo config w/ >16 partitions on 386 would blow O/S.
        400 IF 01<4 AND W>16 THEN F0 = -2: IF J# = "W" AND W>16 THEN 403:
              IF F0 = -1 THEN 405

403 M1# = "INVALID CONFIGURATION FOR THIS CPU.": GOSUB'196: GOTO 1150

4/28/95 MVP system with 64K Memory will not show SF'15 on boot
This will not let you boot. Get message "System has already been configured.
Momentarily. On Line 22 need to change SPACEK=TEST to <61 instead of <62.

Fix 20 Li = 0: IF #PART = 1 AND #TERM = 1 THEN Li = 1: IF C#B = "M" AND SPACEK<61 THEN Li = 0:
        IF C#B = "W" AND SPACEK<1024 THEN Li = 0: IF Li = 0 THEN 24: Li = 0: M# = #STAT(2): ERROR Li = 1
4/28/95 IF LOAD A CONFIGURATION WITH MORE THAN 16 PARTITIONS ON CS TYPE CPU GETS HUNG IN ENDLESS LOOP BETWEEN LINES 2120 & 2050. EDITED LINE 2120 TO PRINT OUT MESSAGE & RETURN TO LOAD CONFIGURATION SCREEN.

FOR 2120 IF U < INT(U) OR U = Ø OR U > UL THEN M15 = "INVALID NUMBER OF PARTITIONS FOR THIS CPU." : GOSUB '196: GOTO 1150

FIX 2120 IF U <> INT(U) OR U = Ø OR U > UL THEN 2122: ELSE GOTO 2125

2122 M15 = "INVALID NUMBER OF PARTITIONS." : GOSUB '196: GOTO 1150

11/4/96 IF SET GENPART FOR AUTO EXECUTION BY PLACING REM IN FRONT OF GOTO 1150 ON LINE 100, STILL PROMPTS FOR "RECONFIGURATION PASSWORD?"

FIX 103 GOTO 425

103 REM GOTO 425: REM %ACTIVATE GOTO TO ELIMINATE PROMPT FOR PASSWORD ON AUTO EXECUTION
INVALID CONFIG ON VLSI

OLD GENPART

F₀ = -2

LINE 400 HAS INVALID CONFIG MESSAGE

F₀ 10 95 130 176 204 340 345 352 380 400 590 1210 1230
     1232 2010 2760 3090

NEW FORMAT GENPART

IF US = " " THEN F₀ = 1  USE = 20 so F₀ = 1
IF CΦS = "W" AND US = " " THEN F₀ = -F₀  COφS = M HD 386 = -1
IF CΦS = "W" AND US = " " THEN F₀ = -F₀  TURBO?

1220 US = US
1232 IF US = " " THEN F₀ = 2  USE = HEX 2D  W₀ = HEX 0D  VLSI 1
1232 IF US = " " THEN F₀ = 1  VLSI 1
1300 Datasave DC #0 F₂B W

1300 Datasave DC #0 F₂B W

Csysfile US CONFIG FILE 20 56,5010 5,550
2 DO = "Csysfile" 16 DO = "Csysfile'

F₀$ 10 16 52 870 940 950 970 980 1210 1340 2310 2320
OLD GENPART
LIST V U$
12 72 95 905 1220 1232 1375 1970 2160 2180 2710 2760 2800
3885

C: GENPART
CHANGE LINE 2760

$M1$ = HEX(00 40 80 CO)
TO " 20 

C$ = #PSTAT(I)
10 DIM C$ = STR(C$)
$M1$ = 2
$X1$ = 2

C:

J$

11 DIM $M1$(I): $M1$ = STR(C$9$, I)
2760 GO$SUB'93$: IF $OL$<$4 AND $X$<2 THEN $M1$ = "": IF $M1$ = "M" THEN 2770
2765 $M1$ = HEX(00 40 80 CO): FOR $X$ = 1 TO 4: GO$SUB'93$(X): NEXT $X$: GOTO 2775
2770 $M1$ = HEX(20 40 80 CO): FOR $X$ = 1 TO 4: GO$SUB'93$(X): NEXT $X$: RETURN
2775 RETURN
OLD FORMAT  12,12,20,2730,2740

SYSFILE

CHANGED BYTE 22 to $EYSFILE from 00 to 20 & WORKS 1st TIME THROUGH BUT WHEN SAVE CONFIG FAILS AGAIN BY CHANGING IT BACK TO 00.

AT INITIAL CONFIG SCREEN  
$U = 20$   
$W9 = 20$   
$W9H = 2020$  

SF 15  
$U = 20$   
$W9 = 20$   
$W9H = 20$  

OK TO EXEC Y  
$U = 20$   
$W9 = 20$   
$W9H = 20$  

RECONFIG PASS WORKS  
$U = 0D$   
$W9 = 20$   
$W9H = 20$  

LIST V $U = 13 \ 72 \ 95 \ 905 \ 1220 \ 1232 \ 1375 \ 1970 \ 2125 \ 2160 \ 2180 \ 2710 \ 2760 \ 2780 \ 2790 \ 2860 \ 3085 \ 9970$  

$E70$ DATALOAD "$EYSFILE"  

$2780 \ U = U9 \ AND \ HEX(1F) \ XOR \ STR(1616/X) \  
HEX PRINT $M16$ 004088C0 2020 0  
$M16$ 12 395 400 639 640 680 1210 1340 1440 1445 2310 
2760 2780 4120 4125 4140 4200 4240 4395 9110
NEW DISK
SCSI BACKUP NO
CBACKUP NO
SCSI RESTORE NO
CRECOVER NO
SCSI CONFIG NO

Failing C Boot
9010 DATA "CBOOT","","6,0"

9080 "CD62", "TURBO", "M"
CD61 "M"

VLSI
DIAGS CAME UP NOT A PROGRAM RECORD
IF BYTE FROM NON-VLSI CONFIG FAILS, MUST ASSERT & RESTART, SHOULD REJECT

1360 PRINT OK CONFIG # PART CPU
1374 IF 01 > 3 AND WH < " THEN Y# = "Y": IF 01 < 4 AND WH = " THEN Y# = "Y"
1375

386 SAYS Y FOR TURBO CONFIG
INVALID CONFIG FOR THIS CPU IF USE VLSI CONFIG
TURBO CONFIG #40 PART ACCEPTED BUT BROKE CPU
DIAGS OK
CREATB CONFIG OK
SAVE OK
\[ W = \# \text{TERMINALS} \]

400

384 ~ LOAD VLS1

395 2000 DMAPN 204 E8 = 23 U = S K = 1

1730 DMAPN 206 I = 1 P = H

1740

1750 BB 2020

1760 J = 1 U = 9

1770 F0 = -1 B = 16 R(J) = 56

1780 I = 1 M = 2 B16 S(J) = 0 X = 0

1785 Z = 0

1790 X8 = 208 BB(J) = Y

1810 E8 = ""

2000 E8 = ""

395 IF COB = W THEN GOSUB 2710

2710 U8 = "" PRINT CRU # IF 386

395 LI = 1

400 D1 = 5 W = 5 FO = -1 M1# = INVALID CONFIG
2020 $A(I) = AVAILABLE MEM$
$S9 = TOTAL SYSMEM$

384 SYS LOAD TURBO
SF 15 HALT 395

0000 GO SUB 204 $E8 = HC 'L'$ $U = 6$ $K = 0$
2040 $E8 = " "$
3000 DEFN' 206 K = 0$
3050 1730 DEFN' 206 I = 1 $P = 0$

1740 \begin{align*}
S(J) &: 62.5 \\
J &: 1 \\
L &: 2 \\
62.5 & \quad \text{Y} \\
C &: 4Y4 \\
P &: SYSTEM
\end{align*}

1750 \begin{align*}
B0 &= 202020 - \\
1760 \\
1770 \quad F0 &= 2 \\
B &= 1 \\
R(J) &= 5692
\end{align*}

1780 \begin{align*}
I &= 1 \\
M &= 2 \\
B &= 1 \\
S(J) &= 62.5 \\
x &= -34823.2
\end{align*}

1810 $E8 = " "$
2000 $E8 = " "$

395

2710 $U8 = 0.5_8$ $U9 = CPU$
395 \text{ LI = 1}$

400 \begin{align*}
O1 &= 5 \\
W &= 40 \\
F0 &= 2
\end{align*}

405 $X8 = " "$
1860 $X8 = "F"$
1870 $J = 0$
1900

1920 \begin{align*}
R(J) &= 8192 \\
S(J) &= 62.5
\end{align*}
R(J) KEEPS TRACK OF MEMORY LEFT

1950
1900
405
415
400 IF O1 = 4 AND W > 16 THEN F0 = -2: IF C0# = "W" AND W > 16 THEN 403: IF F0 > -1 THEN 405

403 M16 = "INVALID CONFIG"
386 SYS LOAD 386X CONFIG

SF '15 PART 395

2000 DEFFN '204 E=320 U=8 K=1
1730 DEFFN '206 I=1 P0=1
1740 S(U)=100 J LA=2 TB=01.02.03.04.05
CB=YY Y - PB=SYSTEM

1750 BA=202020

1760

1770 P0=2 B=1 R(U)=7392
1780 I=1 M=2 B=1 S(J)=100 X=-97767.6
1810

2000

395

2710 U9 = CPU # U4 = O2

395 L1=1

400 O1=5 W=8 F0=2
405

1860 X9 = "E"
1870
1900
1920 BA (1)=N
1930 S (J)=100 SUBTRACTS PART SIZES FROM TOT MEM
1950
1900
405
415
Mike

Change line 5120 of @GENPART as follows:

5120 IF STR(Q2$(I),2,2)=HEX(00) THEN STR(Q2$(I),2,2)=HEX(30 30)
: HEXPACK STR(T2$(I),9,1) FROMSTR(Q2$(I),2,2)
: IF Q3$(I)="0" OR Q3$(I)=HEX(00 00) THEN Q3$(I)="00"
: CONVERT Q3$(I) TOA9
: STR(T2$(I),10,1)=BIN(A9)

Still no ECO for the MB!

Regards

John Baxi
C INSTALL
4/9/93 Added tests for 3 byte addresses, for Input & Output Addresses.

175 DIM A$(1): A$(1) = STR$(S$(1,1,1)): IF A$(1) <> HEX$(02) THEN 177: PRINT AT (22,0,79); HEX$(07,0E): "Address":; A$:; "Is a 3 byte address. Three Byte Addresses Not supported." : GOTO 140

177 PRINT AT (22,0,79)
215: A$(1) = STR$(S$(1,1,1)): IF A$(1) <> HEX$(02) THEN 217: PRINT AT (22,0,79); HEX$(07,0E): "Address":; A$:; "Is a 3 byte address. Three Byte Addresses Not supported." : GOTO 180

217 PRINT AT (22,0,79)

3/4/94 Updated to load CS/386 0.5.3.6 from either 1.2M or 3.360K diskettes.

Changing number of files included with O/S from 36 to 70.

640 REM % FF encountered: F = F + 1: IF C3 < 3 THEN 680: IF F = 3 THEN 680

642 C1# = "CP": IF C3 > 3 AND F = 1 THEN C1# = "CCLOC"

643 IF C3 > 3 AND F = 2 THEN C16 = "CDSCFG6"

650 PRINT AT (17,9,79); PRINT AT (14,9,71); "Mount System Diskette "; F + 1; "in input drive, then hit any key."

655 GOSUB '10 (C1H,1): IF T <> 0 THEN 660: IF F = 1 THEN PRINT AT (13,9,71); "Second disk not mounted."

657 IF F = 2 THEN PRINT AT (13,9,71); "Third diskette not mounted." : GOTO 650

660 RESTORE LINE 850: IF C3 > 3 AND F = 1 THEN RESTORE LINE 882: IF C3 > 3 AND F = 2 THEN RESTORE LINE 888: GOTO 410

670 REM % FF encountered: X = 0: RESTORE LINE 860: GOTO 410

880 REM: CS/386: DATA 70, file list (.deleted .GYSMDAPB, .STARTDSTART, .CMENU, .CMOVE1, .CPSTAT, .CINSTALL) (add .ClASRSJVI & .CHQ39Q0)

Line 890: change to 882 + 884 (deletions from 880 addre) (Additions to 880 lost)

Lines 888 + 890 added for CS/386 Disk 3.
CHANGE MESSAGE ON LINE 280 TO BE MORE DESCRIPTIVE.

280 PRINT HEX(91, 9E, 96); AT(22, 0, 79); R$; "CVP, CMVP - NOT ON INPUT DISK.
INSET CORRECT DISK OR CHANGE ADDRESS." : GOTO 140

3/10/94 UPDATED TO LOAD CS1/TURBO OS 1.30.01 FROM 3.600 Ks OR 1.2M DISKETS.
CHANGING NUMBER OF FILES INCLUDED FROM 65 TO 75.

230 D1$ = "CMVP" : GOSUB '20(D1$) : IF T = 0 THEN 240 : C3 = 3 : E2 = E1 - B : IF E2 >
450 THEN C3 = 4 : IF E2 > 1000 THEN C3 = 5

235 IF C3 <> 4 THEN 237 : PRINT AT(8, 0, 79); "INPUT DISK CONTAINS CS1/TURBO OS."
237 IF C3 <> 5 THEN 240 : PRINT AT(8, 0, 79); "INPUT DISK CONTAINS CS1/TURBO OS."

240 D1$ = "CMVP" : GOSUB '20(D1$) : IF T = 0 THEN 240 : C3 = 3 : E2 = E1 - B : IF E2 >
290 THEN C3 = 4 : IF E2 > 1000 THEN C3 = 5

295 ON C3 : GOTO 300, 310, 320, 325, 327 : PRINT HEX(07) : GOTO 290

325 D1$ = "CS GEN 386" : GOTO 330

327 D1$ = "CS1/TURBO DISKETTE 2" : DATA 920 : END

640 REM %, END ENCOUNTERED: F = F + 1 : IF C3 = 3 THEN 680 : IF F = 3 THEN 680
660 RESTORE LINE 850 : IF C3 = 4 AND F = 1 THEN RESTORE LINE 882 : IF C3 = 5 AND
F = 1 THEN RESTORE LINE 810 : IF C3 = 4 AND F = 2 THEN RESTORE LINE 888 : IF C3 = 5

AND F = 2 THEN RESTORE LINE 930 : GOTO 410

745 ON C3 : GOTO 770, 775, 780, 785, 787 : STOP #
787 RESTORE LINE 900, X + 1: HA = "CS1/TURBO BASIC-2" : C3 = 5 : RETURN

900 REM CS1/TURBO DISKETTE 2 DATA 920 : END
910 REM CS1/TURBO DISKETTE 3 DATA 940 : END
CHANGED COPR DATE, ON MENU HEADING TO 1994, LINE 120.
ADDED UPDATE REMARK, LINE 12.

3/2/95 WHEN RUNNING CINSTALL ON A TURBO, ALWAYS COMES UP WITH CMVP NAME ON SOURCE = C386. ALSO HAD A USELESS PRINT COMMAND.
ON LINE 385 DELETED: D49 = "C386": PRINT:
155.000M  

CHANGES MADE FROM 5.5 TO 5.5+

DELETED CPU RESTRICTION (ORANGE - DELETED CODE)

9/29/89

420 A## = &PSTAT (#PART) IF STR(A##,9,1) = "M" THEN S3 = 4 ELSE S3 = 3: IF S3 = 4 AND STR(A##,10,1) < HEX(17) THEN 1440: IF S3 = 3 AND STR(A##,10,1) < HEX(19) THEN 1460:

#610/D05(7601,A##) AL## = A## AND HEX(10): IF STR(A##,11) = HEX(10) THEN S0 = 80: ELSE S0 = 64: SELECT PRINT D05(50): S = SPACE K

155.205S  

ALLOWS ADDRESSES BEYOND D X 5  

( CHANGES MADE FROM 5.5 TO 5.5+)

9/29/89  

(ORANGE - DELETED CODE)

8.75 D05N'205(R,W3#,R): Q# = ""  "MAT SEARCH" 310320330350360370B10B20B30B50 840B70D11D12D13D14D15D20D21D22D23D24D25D30D31D32D33D34D35D50D51D52D53D54 9D5D6D6D6D6D6D70D71D72D73D74D75 = STR(W3#,3) TO W2#: STEP 3: IF R1 <> 0  AND R1 < > 1 OR R < > 0 OR R > 15 THEN Q# = "X": IF W3# = "340" THEN 8895: HTRAN(W3#,  "AaBcCdDeEeFf") R: IF POS("DB3" = W3#) < > 0 AND POS("123567" = STR(W3#,2)) < > 0  AND VER(STR(W3#,3), "H") < > 0 THEN 8885: "" IF W2# = HEX(9999) THEN Q# = "1":

GOTO 8905  

8885 IF POS("3B" = W3#) < > 0 OR POS("123" = STR(W3#,2)) < > 0 OR STR(W3#,3) < > "0" THEN  

8895: IF STR(W3#,3) = "3" THEN STR(W3#,3) = "1": STR(W3#,1) = "D"

8895 Q# = ""

8905 IF R1 = 0 OR Q# < > "" ""THEN RETURN: SELECT #R < W3#>: RETURN

8915 IF R1 = 0 OR Q# < > "" ""THEN RETURN: SELECT #R < W3#>: RETURN
ADDED SUPPORT FOR CS/386 (CHANGES MADE FROM 5.5 TO 5.5+)

9/29/89

7167 R9A(2) = RPSAT(1): IF STR(R9A(2),9,1) = "Y" THEN 7179: IF STR(R9A(2),9,1) = "W" THEN RETURN: IF STR(R9A(2),10,1) > HEX(24) THEN RETURN: PRINT .......
<table>
<thead>
<tr>
<th>Command</th>
<th>Change</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISS.000A</td>
<td>132-134, 136, 137</td>
<td>(131-139)</td>
</tr>
<tr>
<td>ISS.050S</td>
<td>190-194, 197-200</td>
<td>(189-201)</td>
</tr>
<tr>
<td>ISS.001M</td>
<td>205-207, 210, 213-218</td>
<td>(204-219)</td>
</tr>
<tr>
<td>ISS.002M</td>
<td>514-524</td>
<td>(513-525)</td>
</tr>
<tr>
<td>ISS.081U</td>
<td>540, 553</td>
<td>(539-551)</td>
</tr>
<tr>
<td>ISS.090U</td>
<td>575, 579</td>
<td>(574-582)</td>
</tr>
<tr>
<td>ISS.REF1</td>
<td>644-647</td>
<td>(644-649)</td>
</tr>
<tr>
<td>ISS.205S</td>
<td>651-654</td>
<td>(650-655)</td>
</tr>
<tr>
<td>ISS.001U</td>
<td>672-683-692</td>
<td>(671-692)</td>
</tr>
<tr>
<td>ISS.000M</td>
<td>820-830, 833</td>
<td>(819-832)</td>
</tr>
<tr>
<td>ISS.000E</td>
<td>872</td>
<td>(870-872)</td>
</tr>
<tr>
<td>ISS.012U</td>
<td>853, 867</td>
<td>(853-867)</td>
</tr>
</tbody>
</table>

**Change in Rel 5.5+**

2° Second sector on all other files was different. Appears to be due to version. All files on Baxi's version appear to be version 5.3 or 5.2, otherwise same as Rel 5.5.

**Baxi Changes:** Shows & allows all disk addresses. Recognizes Turbo.
4/7/93  Added test for 3 byte index for input & output addresses. Message comes up "3 byte addressing not supported."

50 DIM A$(1)
150 PRINT: INPUT: IF: IF: C1 = : IF: PRINT: C1: GOSUB 105: A$(1) = STR$(S$(1),1,1): IF A$(1) = HEX(02) THEN 155
152 IF QB = "" THEN 160: A3$ = "": GOTO 130
155 IF A$(1) <> HEX(02) THEN 160: PRINT AT (14,10): HEX(0E): "Address " : A1$: " is a 3 byte address. Three byte addresses not supported." : GOTO 130
140 GOSUB 105(2,A2$,C2): A$(1) = STR$(S$(1),1,1): IF A$(1) = HEX(02) THEN 1445: IF QB = "" THEN 1450: A5$ = "": GOTO 410
1445 IF A$(1) <> HEX(02) THEN 1450: PRINT AT (14,10): HEX(0E): "Address " : A2$: " is a 3 byte address. Three byte addresses not supported." : GOTO 410
Move File Utility fails w/ message "Output Platter Full" if 60 becomes 65535.

Input D10 Output D34 Move All Y Overwrite N HALT|STEP

142 HA is OVERWRITE Y/N

200 A1# = D10 C1 = 0 Q$ = 20H

5140 DEFFN 115 N S SH T

5142 RETURN

00\$1 (SH(1),1) IF RS(1) = HEX(0) THEN CHECK INDEX TYPE

200 I1 = VAL (STR(SH(1), 2)) I1 IS INDEX SIZE OF INPUT DISK

210

220 GOSUB 130 (0, 1, C3, A9, I1-1, A1#, C1):

5180 DEFFN 130 (0, N, NB, T8, T9, SH, T) = N3H = ALL(00) Q$ = N = R3 = 0 I = T8 TO T9

5140 DEFFN 115 (N, S, SH, T) : = DATALOAD BAT\#N, (S) SH(

5142

5180 FOR J = 1 TO 16: IF POS(SH(J, 1) <> HEX(0)) = 0 THEN 5220

IF I = 0 & J = 1 OR STR(SH(J, 1), 1) = HEX(1) THEN 5280

5230 NEXT J, I:

5180

5190 N3B = IF POS(NA = HEX(0)) = 0 OR 0 < 0 THEN 5200

5200 R3 = -SGN(V(VAL(STR(SH(J, 1), 2), 1))) + 2:

STR(SH(J, 1)) = 1000\$1 0001 0002 0003 0004 0005 FILE ENTRY 24 103

IF \$1 = HEX(1) THEN R3 = -R3 (IS FILE SCRATCHED)

R4 = VAL (STR(SH(J, 1), 3), 2) R4 = 24 START SECTOR

R5 = VAL (STR(SH(J, 1), 2), 2) R5 = 103 END SECTOR

R1 = 5: R2 = J

5205 GOSUB 115 (N, RS, SH, T)

5140 DEFFN 115 (N, S, SH, T) DATALOAD BAT\#N, (S) SH(

5142
5205  R6 = VAL(STR(SB(1),Z),Z)  READ 8 BYTES 2 & 3 OF USED LAST SECTOR FOR USBR FIELD
5210  Q$ = " " : SB = J  S$ = I  I = T9
5220  J = 18
5230  NEXT J, I
220  IF N3$ = " ESPAN01" THEN C1B = " Y" : IF QA = " N" + C9$ = " Y" THEN 250
240  Z$ = C1B, C2$ = C3$ = N3$
242  R9 = S9
270  P9$ = C1B : IF Z = 1 THEN 280
280  C6$ = C1B
290  $ OPEN #1, #2 "GETTING FILE PARAMETERS" : IF C1B = N3$ OR QA = " " OR P9$ = C1B THEN GOSUB 130(1,1,C1B,O,11-1,A1H,C1)
5180  DEFN' 130(0,N,NB,T8,T9,SB,T)
5180  N3$ = " O": QA = " N": R3 = φ : 1 = T8 TO T9
5190  GOSUB 115(N,I,SB,T)
5140  DEFN' 115(N,S,SB,T)  DATA LOAD BAT#N,(S)S$()  LOAD SECTOR φ
5142  RET
5180  FOR J = 1 TO 18  IF I = φ & J = 1 : 5230
5230  NEXT J, I
5180  N3$ = S$(J,2)
5200  R3 = - SGN(VAL(STR(SB(J,1),2,1)))/128) + 2
5210  IF STR(SB(J,1)), 1 = HEX(1) THEN R3 = -R3  IS FILE SCRATCHED
5210  R4 = VAL(STR(SB(J,1),3),2)  R4 IS START SECTOR FOR C'MX5φ
5210  R5 = VAL(STR(SB(J,1),5),2)  R5 IS END SECTOR FOR C'MX5φ
5210  R1 = S$  R2 = J
MXE Chronological Changes

MVP 2.7/MXE 2.66  July 1986
FIXED
1. **$BREAK** now works properly.
   ANOMALY: HALT/STOP of a LINPUT command may not properly clear display after return.

MVP 3.0/MXE 3.0  July 1987
FIXED
1. Download of MXE code during Master INIT could fail if MXE file contained data beyond trailer record.
2. **$BREAK!** to suspend a partition could inadvertently fail if the **$BREAK** was executed shortly after a KEYIN statement with 2 line numbers.
3. Underlines in a LINPUT field wrapping from 1 line to next would sometimes remain after exit of LINPUT.

MVP 3.1/MXE 3.0L  Sept 1987
FIXED
1. In TC mode, if sending 5 or 6 bit characters, sometimes a shifted character would not be sent until another character was sent.
2. In some instances, printing to address 204 of a 2436WP would stop and would need to be rebooted to correct.

MVP 3.2.7/MXE 3.0A  Slowness on LIST to 204 with DRIVER ON

MVP 3.3/MXE 3.0XX  Feb 1989
FIXED
1. System hesitated up to 1 sec when typing in before displaying & then would display in best.
2. After INPUT SCREEN, system could lose characters if rapidly entered on a KEYIN.
3. System clock would lose time if printing to port 4's local printer on MXE 1.
ENHANCE: Performance similar to MXD. 19.2 baud allowed for TC

MVP 3.4/MXE 3.10  May 1990
FIXED
1. System hesitates printing to address 204.
Enhancement: Performance similar to MXD, allows 38.4 baud.
CPSTAT

ADDED LINE 47 TO DISPLAY ON SCREEN:
PRESS SF'12/NEXT FOR NEXT SCREEN, SF'13/PREV FOR PREVIOUS SCREEN.

3/21/95 FAILED ON NON-386 CPUs w/ ERROR 519. Pointing at #CPU following CONVERT.
FIX: 28 'REM % CONVERT #CPU TO R$,('##)';
Package Subject: Bugs with SCSI Config

Item Title: Bugs with SCSI Config

John,

Found a couple of minor bugs with the SCSI Config program. Sorry to bother you with this but if you still are working with it they are as follows:

1. Ran the SCSI Config program through twice while reviewing the Utility Manual. On the 2nd pass after creating a new configuration, the system put the password entered in the 1st configuration on the dotted line where the number of sectors for surface 1 would be entered. The password was ‘SYSTEM’. It looked something like this:

```
SETUP SCSI CONFIGURATION
```

SECTORS REMAINING : 620178
AMOUNT FOR SURFACE 1 SYSTEM

2. As I mentioned on the phone the system will not accept odd numbers. It just beeps at you. 11/19/92 changed line 1280 which prints out sector prompt to:

```
"SECTORS TO ASSIGN (EVEN # ONLY) TO ADDRESS "
```

3. If last entry when assigning sectors to surfaces exceeds available space the system jumps back to the screen where it asks if you want a Master or Slave. Should allow you to correct your entry. 11/19/92 added test to line 1300 to determine if # exceeds REMAINING SECTORS. IF BOTH THEN 1315...MORE GOSUB 1332 ON END OF LINE 1220 TO NEW LINE 1315. ADD PRINT AT (33/92) TO LINE 1280 TO CLEAR ERROR MESSSAGE AFTER EACH ENTRY. CONNECTED ERROR MESSAGE TO "AMOUNT EXCEEDS AVAILABLE SECTORS.

4. If you enter ‘NO’ when asked if all entries made are acceptable you are sent back to where the system asks you the ‘number of surfaces’ again. That is fine but when you get to the point where you start assigning sectors to each address, available sectors equals 0 and any entry takes you back to problem 3. The utility needs to show the total disk space for that drive again.

```
11/19/92 reset AVAILABLE SECTORS back to total available on drive by adding to line 1410; T1=T.
```

5. As I mentioned on the phone I think it is important to show the drive ID # on the screen where each address is displayed along with the catalog info. It also appears that any drive already configured may be damaged if a new configuration is created even if the addresses don’t conflict. After I created the new configuration I had some strange information showing on the screen showing the catalog information. This could become a major problem when replacing 1 bad drive for a customer with more than 1 drive. Some of the catalogs look like they may be ok but it seems to be different from before. For example 2 of the surfaces had an index size followed by an &: 30& and 3& Is that legitimate? If I don’t hear back from you I will give you a call later in the week.

Best regards,
Mike

11/19/92 change line 1230 to read, "Please Enter No. of Addresses to Assign to Drive.".

12/4/92 added "fn|tab=Exit" to new screen. See line 1029, 1165, 1225, 1440
`Added Warning to apply screen. Line 1440.

12/17/92 change goto on end of line 1410 from 1170 to 1150 to remove all previous sectors values from addresses should you respond No to "Configuration Acceptable?"

Made entry < 100 unacceptable for sector 128. Line 1290 added. If B<100 THEN 1281.

12/21/92 remove & default on all entries acceptable. Line 1390.
12/22/92  IF 0 ENTERED AS SECTOR VALUE ASKS YOU IF ENTRIES ACCEPTABLE.
   LINE 1290  ADDED  IF B = 0 THEN 1340.
   CHANGED LENGTH OF BH TO 1 CHAR. LINE 1280.
   EXTENDED MESSAGE WHEN ASKS FOR CONFIGURATION FILE NAME. LINE 1540.

2/4/93  CONFIGURATION SPELLED WRONG ON LINES 160 & 350, CORRECTED.
PRELIMINARY CHANGE

3/1993 If you have assigned all available sectors but still have addresses remaining that were indicated to be used, \( \emptyset \) would not be accepted. Had to key FN\|TAB to exit and restart.

Fix - If \( \emptyset \) sectors is entered, will end sector assignment procedure and ask if "All entries made and acceptable?"

Remove IF LEN(B\#)<1 THEN 1280 from end of line 1280
Add 1280 : IF B=\( \emptyset \) THEN 1390

Address spelt w/two Aa's (Address) on line 670. Corrected.

3/10/93 If save config & rerun setup, password shows up when on sector assignment field.

1520 GOSUB 602('""""'): B\# = ALL(20): PRINT

- SYSTEM is password. Changed to make SYSTEM the password.

Add 100 DIM V3B

Change from 1510 IF B\#="SYSTEM" THEN 1520: PRINT HEX(07);: GOTO 1500

1510 V3B=DATE: DATE = V3B: PASSWORD B\#: ERROR PRINT HEX(07);: GOTO 1500

3/14/93 If more than 15 addresses are assigned, the first slave address is lost and all address after it move up 1.


- INT X71 on line 660 or 760. Print image not big enough for S. Added 1 #

\( \rightarrow \) Converts to L8 on lines 660, 670, 720, 760
3/11/93 When expand display catalog maximum field drops least significant field. 65024 show as 6502. Append # to left of catalog max field on line 7920.
LOAD CONFIG M OR S M
ADDR'S 29
SECTORS 1-F 60000
71 50000
72 40000

ADDR 15 60000 HALT/GET

1280 PRINT
1290 CONVERT B6 TO B IF = 0 END, IF <100 REPEAT
1300 C=B MAKE SURE SECTOR # DOES NOT EXCEED REMAINING + 1 IS ODD
1320 S2=SECTORS USED + 2 (GOSUB 84000) 84000Z 14 X 60000 = 84000
8070 DEFN'600(0) D=S2 84002
8080 GB=BIN(P,4):FB=ALL(20) G$=0000CD142 FB=60EA0000
8090 STR:F$=GB REVERSE
1320 STR(A$(1),P,4)=FB
    P=117
    P=P+4 P=121
GOSUB'600(C) C=60000

8070 DEFN'600(0) D=60000
8080 GB=BIT(P,4):FB=ALL(20) G$=0000EA60 FB=42D10C00
8090 STR:F$=GB REVERSE
1320 STR(A$(1),P,4)=FB
    P=121
    S2=S2+C UPDATE SECTORS USED S2=900002 C=60000
    B=60000 X=15
    T1=T1-B UPDATE REMAINING SECTORS

1330 IF X<15 THEN DO: P=P+4: END0 X=15 P=121
1340 IF X=15 THEN DO: P=P+4 P=125
    STR(A$(1),P,4)=HEX(FF FF FF FF)
    P=P+4 P=129
    STR(A$(1),P,3)="END": P=149: END DO P=149
    *** P=153

1350 IF X=16 THEN DO
1360 IF X>16
1370 IF X=0 THEN DO PRINT FIELDS R2=8 A=7 X=15
1380 NEXT X
1270 PRINT "SECTORS REMAINING" TI 426202
1280 INPUT SECTORS FOR ADDR 16 50000
1290 CONVERT B$ TO B, IF @ END, IF <100 REPEAT
1300 C = B MAKE SURE DOES NOT EXCEED REMAINING SECTOR & NOT ODD
1320 (C = 50000) (T = 1326202) (A(X) = 50000) (S2 = 900002) GOSUB
8070 DEFFN'600 (D) (D = 900002)
8080 G$ = 0000DBBAZ F$ = 60EA0000 CLR F$
8090 STR F$ = G$ REVERSE G$ = 0000DBBAZ F$ = A2BB8DD4
1320 STR(A$(C), P, H) = FA (P = 149)
P = P + 4 (P = 153) GOSUB'600(C) C = 50000
8070 DEFFN'600 (D) D = 50000
8080 G$ = 0000C350 F$ = A2BB8DD4
8090 G$ = 0000C350 F$ = 50C38DD4
1320 STR (A$(C), P, H) = FA (P = 153)
S2 = S2 + C UPDATE SECTORS USED
B(x) = S2 B = 50000 X = 16
TI = TI - B UPDATE REMAINING SECTORS

NO
1330 IF X < 15
1340 IF X = 15
YES
1350 IF X = 16 (P = 149) V = 149
STR (A$(C), P, H) = "SG8"
V
P = P + 4 (P = 153) V
R2 = 48

NO
1360 IF X > 16
1370 IF X > 17 THEN DO PRINT HELPS R2 = 48 A = 7 X = 16
1380 NEXT X
1270 PRINT SECTORS REMAINING 316202
1280 INPUT SECTORS FOR ADDR 17 40000
1290 CONVERT B$ TO B, IF @ END, IF <100 REPEAT
1300 C = B C = 40000
1320 C40000 + 1326202 A(X) = 40000 A = 7 X = 17 GOSUB(S2) S2 = 95002
8070 DEFFN'600 (D) D = 95002
8080 G$ = 000075E2 F$ = 50C38000
8090 FA = GA REV
8094 GB = 00DE7EF2 FB = F27E6E64
1320 STRA$( )', P, 4$ = F#$ (P = 153)
P = P + 4 (P = 157)
GOSUB '600(C) C = 40000
8020 DEFFN '600(P) D = 40000
8080 GA = 00009c40 FB = F27E6E64
8090 GB = 00009c40 FB = 409C0000
1320 STRA$( )', P, 4$ = F#$ (P = 157)
S2 = S2 + C UPDATE SECTORS USED S2 = 990002 C = 40000
B(X) = S2 G = 40000 X = 17
T1 = T1 - B UPDATE REMAINING SECTORS
1330 IF X < 15
1340 IF X = 15
1350 IF X = 16
1360 IF X > 16
P = P + 4 (P = 161)
1370 PRINT
1380 NEXT X
1270 PRINT SECTORS REMAINING
1280 INPUT SECTORS FOR ADDR 18 30000
8000 REM 16: Apply Changes to DS cabinet: REM #. Code # B000-8990 will be a separate module saved scrambled as 80C88F88
8010 DEFDB 'X'; IF #TERM<>"1" THEN 10: IF#2=SELECT #2: STR(0#,2,1)=STR(0#,2,1) AND HEX(0#): STR(0#,3,1)="0": D1#=SELECT #1: PRINT HEX(03): PRINT AT(0,0): "ALL DATA ON A INCHES is in THE UNIT BEING CONFIGURED SHOULD BE BACKED UP!!" 8012 PRINT AT(23,35):"FM/TAPE - Exit": AT(1,0): "Apply changes to cabinet ": D1: D FROM file on ":D1"
8015 X=4: IF STR(E#(1),4,2)>"3": THEN B020: GO SUB '50(HEx(1", "You need DS proc level 4 to apply changes'": KEYIN KM: GOTO 10
8020 PRINT AT(2,0):HEX(0E): "Enter PASSWORD to apply changes": X=": PRINT AT(2,40): LINPUT =X: V3=DATE: DATE=V3$PASS$02#: ERROR PRINT HEX(07): X=X-1: IF X<0 THEN B020: GOTO 10
8025 F#="$DEFAULT"
8030 PRINT HEX(07): PRINT AT(2,0,80): LINPUT "Configuration file name ":F#.
8040 LIMITS T#1,F#1,W,1,E: IF E>2 THEN B030: IF E<6 THEN B030
8045 PRINT AT(1,0,80):HEX(0E): "APPLYING UPDATE TO DS PR 0 M": D1=1
8050 REM 1: Drive configuration sector: REM *Bytes.001-032 Drive constants: REM . 
001-007 Reserved for alternate sectors: REM .008-010 reserved for future use.
"REM .011-013 "TBO": REM .014 binary count of platters in drive: REM .015 st 
platter eq 01, 05, 41
8060 REM 0.016heads/drives REM .017 cylinder to start RWC: REM .018 # cylinder 
s for alternate sectors: REM .019 drive switch setting: REM .020-032 reserved fo ;
3/15/93 If type in wrong ADR# or a non-SCSI addr, or no tape drive is found, FN/TAB message to exit disappears.

Fix: Change GOTO 150 on lines 150, 180, 2185 to GOTO 130.
CSCTAPER

3/15/93 If type in wrong address for tape, a non-SCSI controller error, on tape is not recognized, get error but FN/TAB message to exit is gone.
Fix: Change GOTO90 on lines 90, 120, 125 to GOTO 70.

3/16/93 FN/TAB disappears when tape directory loaded.
Fix: Change PRINTAT statements on lines 7040-7060 from (230,79) to (23,0,60).
3/4/6 MVPB

2/20/95 Added new line
9115 DATA "Clock", "Display Clock", "P"
12/23/92 Change screen heading: Line 35
From: Creating | Moving a Selected List of Files - (C) COPRA, WANG LABS...
Removed X from 4 save options. Lines 521, 522, 523.

4/7/93 Added test for 3 byte index for input address. Not supported.
55 VERIFY T#2 (0,0) A: ERROR A = 1: IF A = 1 THEN 60
57 DIM B$(16), A$(1): DATALOAD BA T#2, (0) B$( ): A$(1) = STR (B$(1), ,1): IF A$(1) <> HEX(02) THEN 60: PRINT AT (2,25); HEX (0E);" Address " ; D$: " is a 3 byte address. Not supported. " : GOTO 50

6/8/93 Changed line 55 to circumvent Turbo bug where VERIFY T(0,0) to 2275 verifies whole disk & returns 198 error.
55 VERIFY T#2, (0,1) A: ERROR A = 1: IF A = 1 THEN 60
CS/2200 Tool Box Option Updated
by Tyler B. Olsen

[Editor’s note: Tyler Olsen is the primary author of the CS/2200 Tool Box Option.]

An updated version of the CS/2200 Tool Box Option, offered through the ISWU Software Library, is now available. The Tool Box Option is a subset of a collection of BASIC-2 programs that have been developed over the lifespan of the CS/2200 line to facilitate the development and maintenance of system software.

These programs are collected and maintained in a loose-leaf notebook. The programs are created by a variety of people but are collected and sometimes modified by one.

This update includes several changes from earlier versions. The cross-reference lister has been modified to include most BASIC-2 release 3 language enhancements. Disk prompting has been modified to include addresses of the Data Storage Cabinet. New features have been added to the disk catalog sort. Other program functions have been added, some enhanced.

TOOL BOX GROUND RULES

My ground rules for developing and maintaining this collection are:

- The programs must reside on single diskette surface. The maximum number of sectors is 1232 on a 2270A-type drive. These criteria keep the number of programs to a manageable number.
- The basic operating instructions are to mount the diskette, then key RESET LOAD RUN RETURN and follow the prompts. There is documentation with the package which provides program abstracts in addition to operating instructions and descriptions of some functions.
- The functions must be useful to the CS/2200 system user. System diagnostics enable you to look at the hardware to see what is connected. Interrogation tools enable you to look at programs and data files. Documentation aids provide helpful tools for development or system maintenance. And some of the programs are for the creation, listing, and maintenance of telecommunications-formatted data files.

Tool Box Option diagnostic programs include “Monitor Partition Status,” “What Disks Are On the System?” “Show Status Of Printers, TC Boards, and Terminal,” “Analyze Disk Index,” and “Show CRT Character Set and Keyboard Values.” Interrogation tools include “Sort the Disk Catalog Area,” “Disk Catalog Examination,” and “File ZAP.”

Documentation and development aids include “Sort the Disk Catalog Area,” “Program or System Comparison,” and “Cross-Reference Listing.”

Other programs represent experiments that might be useful or of interest; e.g., “Analyze Disk Index,” “Flow Chart Maker,” “Data File Comparison,” “Number Conversions,” “Analyze $GIO Statements,” “@CLOCK” a clock and calendar, and many others.

Some of the programs contain one or more subroutines to perform a particular function. You may want to study these routines for code that could easily be included in your own system software. TBODISKS, for example, has code to look for all possible disk addresses; TBO.XDAD is used to determine the disk type; and @CLOCK creates large block letters and digits on the CRT.

If I find myself on an unknown system, I might run “Monitor Partition Status,” “Show Status of Printers,” and “What Disks Are On the System?”
If I'm looking at software packages, I am liable to use "Sort the Disk Catalog Area," "Cross-Reference Listing," or "Program or System Comparison." If used with all their bells and whistles, these three functions are adequate to document and maintain a software system.

A FEW PROGRAM DETAILS

The "Cross-Reference Listing" creates a cross-referenced listing of one or more programs from one or more disk surfaces. This program has been upgraded to process most of the language enhancements of release 3 of the Wang Multi-User BASIC-2 operating system.

The cross-reference program creates a listing of each program module that is page-numbered and titled. The first section is a decompressed listing of each program line; the summary section describes file references and variables used, prime subroutine locations, then statement and prime subroutine references.

The program uses tables that enable the user to describe BASIC-2 variables, prime functions, and file references with mnemonic descriptors of up to 16 characters. These mnemonic descriptors may be output in the margin of the listing as well as in the summary description. Use of the mnemonic descriptors is optional, but they can be extremely useful in documenting programs and systems.

"@CLOCK" uses the MXE board to provide a data and time clock. The clock has a large digital face and monthly calendar or message display area. A small file of operator-entered reminder messages is maintained for each terminal.

"Sort the Disk Catalog" sorts the items in the catalog index area for viewing. Data or program file names may be ordered by sector number, reverse sector number, or name. If ordered by name, either all or a common one- to eight-character ID may be requested. The display will show name and sector information only as with LIST DCT, but sorted. Alternatively, it will show the sector information with the descriptive remark or the image statement normally found as the first line of any module. Provision is also made for output of descriptions of data files.

The sort has an option of looking for programs modified after a certain date. (I put a MM/DD/YY stamp on the front of each program module.)

Another option lets you look only for the presence or absence of an item from a list of names; this feature lets you see if all modules of a particular system reside on a particular surface. The list of names found is saved in a common array, which may be used as input by other programs. An exit is provided for this feature.

"Program or System Compare" compares two programs line by line and displays the differences. Alternatively, it will compare the modules on two different disk surfaces from a list of program names.

"What Disks Are On the System?" shows the cabinet type for each of the three disk controllers. It also shows the catalog index summary for each disk drive.

ORDERING INFORMATION

The CS/2200 Tool Box Option is available to all ISWU members. The package costs $45 and is available on three types of diskette: 8-inch SSSD, 8-inch DSSD, and 5 1/4-inch.

To order the CS/2200 Tool Box Option (or if you have any questions), call the ISWU Software Library at (617) 967-1058. Have your ISWU membership number as well as a purchase order number ready. To order by mail, send a completed order form (following this article) along with a check or purchase order. Indicate the type of media you prefer on your order.

Allow three to four weeks delivery time for any software order. Express mail service is available for an additional $25. Specify express mail service on your order if you wish to take advantage of it.

Tyler Olsen
is principal software engineer for the Wang Laboratories, Inc. (Lowell, MA) CS/2200 Product Group.
CS/2200 Tool Box Option

What do you wish to do?

TO OPERATE -- Press S.F. KEY or DIGIT corresponding to name, or position # via Alpha, RETURN, SPACE, or BACKSPACE and key enter

- Tool Box Utilities and program development aids
- Clock and calendar
- Tool Box Diagnostic aids
- LOAD RUN xx from another surface
- Monitor Partition status
- Prepare or list TC format data
- @MENU

Tool Box Utilities

- Sort the Disk catalog area
- Cross-reference listing
- Program or system comparison
- File ZAP
- Analyze data file
- Data file comparison

Tool Box Diagnostics

- What disks are on the system?
- Monitor LIST DT
- Scan disks on the system
- CRT character set and keyboard values
- Show status of Printers, TC boards, & terminal
- Analyze $GIO statements
- Analyze Disk index
Wang Micro-VP Tool Box Option Software

The "Wang Tool Box Option Software" is a loose-leaf collection of Wang 2200 Basic-2 programs that were developed to aid and assist in the development of Wang 2200 systems. The programs may be accessed from a simple command menu.

The 2200 Technical Support group has used these programs and thought they might be useful to the field as well.

Programs are Unsupported

Programs are a Loose-leaf collection subject to change

Documentation is Reset, LOAD, RUN, RETURN

and read the prompts on the CRT

The "Wang Tool Box Option Software" diskette includes:

Utilities and Program Development Aids

eg.

Cross-reference and Sort catalog area

Clock and calendar

Diagnostic Aids

Monitor Partition status

Prepare or list TC format data

Menu entry to system @MENU

An installation program to move files to a system platter
Wang micro-VP Tool Box Option Software

Index:

Section I: Overview:

Disclaimer

Initial menus

Menu and system files of DATA statements.

Section II: Abstracts for Tool Box Option Software:

Section III. Operating instructions for selected utilities.
Wang Micro-VP Tool Box Option Software

The "Wang 2200 Tool Box Option Software" is a loose-leaf collection of Wang 2200 Basic-2 programs that were developed to aid and assist in the development of Wang 2200 systems. The programs may be accessed from a simple command menu.

Tool Box Option MM/DD/YY What do you wish to do?
TO OPERATE — Press S.F. KEY or DIGIT corresponding to name, or position # via Alpha, RETURN, SPACE or BACKSPACE and key RUN

# 1 to Tool Box Utilities and program development aids
   2 to Clock and calendar
   3 to Tool Box Diagnostic Aids
   4 to LOAD RUN xx from another surface
   5 to Monitor Partition status
   6 to Prepare or list TC format data
   7 to @MENU interaction

ToolBox Option Utilities MM/DD/YY What do you wish to do?
TO OPERATE — Press S.F. KEY or DIGIT corresponding to name, or position # via Alpha, RETURN, SPACE or BACKSPACE and key RUN

# 1 to Sort the disk catalog area
   2 to Cross-reference listing
   3 to Program or system compare
   4 to @MOVEFIL disk to disk file copy
   5 to File Z A P
   6 to Analyze data file structure
   7 to Disk catalog examination
   8 to Search Programs for verbs
   9 to Flow chart maker
  10 to Rename data file
  11 to Map disk for program-call integrity
  12 to Data file comparison
  13 to Notes on Tool Box utilities
  14 to Number conversions

Tool Box Diagnostics MM/DD/YY What do you wish to do?
TO OPERATE — Press S.F. KEY or DIGIT corresponding to name, or position # via Alpha, RETURN, SPACE or BACKSPACE and key RUN

# 1 to What disks are on the system?
   2 to Scan disks on the system
   3 to CRT character set and keyboard values
   4 to Show status of Printers, TC boards, & Terminal
   5 to Examine ASKACALL file

Installation

The "Tool Box Option" diskette is provided on a single surface. An installation program "TB0.INST" may be used to move the files to a system platter. The entry program is "START" which calls "STARTTB0".
The menus activated by the utility "STARTTBO" operate off a list of DATA statements which normally overlay the code from 9000 – end. These DATA statements are of the form:

Bytes 1-8 Program module name
Bytes 9 value = space if loading a program
       = l if loading a menu overlay (new code for statements 9000-end).
       = c if saving COMMON variables, eg. disk addresses.
       = @ if loading the standard @MENU program.
Bytes 10-70 are descriptive text to appear on menu line.

Line 15 of module "TBO UTIL" allows you to specify the most common disk surfaces on your system. When asked to specify a specific surface the operator may then key a single digit or a full disk address and EXEC.

0015 COM D1$(12)4; D1$(12)="310 B10 320 B20":REM/.up to 9 disks

9000 REM .......V1.......2.......3.......4.......5.......6.......
9... DATA "Tool Box Option"
9... DATA "TBO UTIL" TBO Utilities
9... DATA "@CLOCK Clock and calendar
9... DATA "TBO DIAG" Diagnostic Aids
9... DATA "TBO TCPL" Prepare or list TC format data
9... DATA "@SYSMVPB@MENU"
9... DATA ""

Within the menu "TBO UTIL"
9000 REM TBO UTIL
9... DATA "Tool Box Option Utilities"
9... DATA "TBO.SDCcSort the disk catalog area
9... DATA "TBO.CRFcCross-reference listing
9... DATA "TBO.CMPScProgram or system comparison
9... DATA "@MOVEFIL @MOVEFIL"
9... DATA "TBO.ZAP File Zap
9... DATA "TBO.ANDFcAnalyze data file structure
9... DATA "TBO.XDC cDisk catalog examination
9... DATA "TBO.SPV csSearch programs for verbs
9... DATA "TBO.FLOWcFlow chart maker
9... DATA "TBO.RENMcRename data file
9... DATA "TBO.DMAFcMap disk for program call integrity
9... DATA "TBO.CMD csData file comparison
9... DATA "TBO NOTE Notes on TBO utilities
9... DATA "TBO NUMc]Number conversions
9... DATA ""
9... DATA "TBO.CRF1 Crossref COMMON
9... DATA "TBO.CRF2 Crossref set-up
9... DATA "TBO.CRF3 Crossref mainline
9... DATA "TBO.SDC1 SORTCAT mainline
9... DATA ""

Within the menu "TBO DIAG"
9000 REM TBO DIAG
9... DATA "Tool Box Diagnostics"
9... DATA "TBO.XDAD What disks are on the system?
9... DATA "TBO.CRT CRt character set and keyboard values
9... DATA "TBO.STAT Show status of Printers, TC boards, & terminal
9... DATA "TBO.XASK Examine ASKACALL file
9... DATA ""
II. Abstracts for Tool Box Option Utilities

Function: Cross-reference listing
Modules: TBO.CRF0 - TBO.CRF1 - TBO.CRF2 - TBO.CRF3
Abstract: One or more program files on any system disk may be listed on numbered and titled pages. For each program requested the listing consists of decompressed program statements followed by a cross-referenced listing of variables used, special function, and numbered statement references.

BASIC-2 commands operate on a program in the user partition:
LIST lists all lines.
LISTSD lists a Section Decompressed.
LIST V lists Variables used.
LIST # lists Special Function references.
LIST S lists numbered statement references.

Function: Sort the disk catalog area
Modules: TBO.SDC0 - TBO.SDC1
Abstract: Sorts the entries of the disk catalog into an ordered sequence. The listing may be in name sequence or in sector number, or reverse sector number order. Program or data file names are listed item by item in separate groups. Listings requested by name may include all or a subset of the files stored.
BASIC-2 command LIST S DCT lists all file names in hash sequence order.

Function: Program or system comparison
Modules: TBO.CMPS
Abstract: Program files on any system disks are compared line by line. Differences between the two files are displayed line by line. Two system disks may be compared by activating a list of DATA statement names.
BASIC-2 commands — None.

Function: @MOVEFIL disk to disk file copy
Modules: @MOVEFIL
Abstract: This program from the standard BASIC-2 release copies program and data files from one disk surface to another.

Function: File ZAP
Modules: TBO.ZAP
Abstract: A single disk surface may be examined. The display shows the contents of a single sector in hexadecimal notation and in ASCII.
1. Items in the catalog index area are flagged as AP (Active Programs), AData (Active Data).
2. Data files are displayed with highlighted attribute bytes.
3. Program files are displayed with highlighted RETURN codes.
Caution, you have the ability to change any all data in a selected sector.
BASIC-2 commands — None.
Function: Analyze data file structure
Modules: TBO.ANDF
Abstract: This program analyzes a catalogued data file for structure. The output is a summary of the SAVE statements used to create the file.
Example
File name = SYSFILE
File size = 32 Sectors.
File = SYSFILE BASIC-2 data structure — SAVE number 1
$8 Scalar $32 $(2)16 $(16)8 $(33)3 $(15)13

Example
File name = BSC*010A
File size = 40 Sectors.
File = BSC*010A BASIC-2 data structure — SAVE number 1
$(256)16
File = BSC*010A BASIC-2 data structure — SAVE number 2
$(256)16
END OF FILE

Function: Disk catalog examination
Modules: TBO.XDC
Abstract: This program lets the operator 1) examine the disk catalog, 2), examine a disk, or 3). Search disk for programs. I use 3) to locate programs by sector number when a disk index has been clobbered.

Function: Search programs for verbs
Modules: TBO.ANDF
Abstract: This program lets you search a list of programs for specific verbs. It was originally written to determine which kind of system was required to run a certain program. I have used it in two different environments:
1) to search systems for COM statements when trying to reduce memory requirements,
2) for places where string variables are set to quotable values, (useful when translating English menus to Spanish).

Function: Flow chart maker
Modules: TBO.FLOW
Abstract: This program was an early attempt to make flow charts from BASIC-2 code. It draws boxes around BASIC-2 code. Who cares? Someone who is required to submit flow diagrams of their program. The cross-reference program when used with REM statements and variable annotation is far superior.

Function: Rename data file
Modules: TBO.RENM
Abstract: This program allows the operator to change a data file name.
Function: Map disk for program call integrity
Modules: TBO.DMAP
Abstract: This program traces the program map of a disk by keying off the program names and descriptions in load modules. In a TBO menu structured disk it will show which menu picks are further menus, which are programs, and which dead-ended", i.e. programs that are non-existent on the disk.

Function: Data File comparison
Modules: TBO.CMD
Abstract: Data files on any system disks are compared byte by byte. Differences between the two files are highlighted and displayed sector by sector.

BASIC-2 commands -- None.

Function: Partition status
Modules: TBO.PSTAT
Abstract: This program built from the SYSTEM UTILITY "@PSTAT" also shows the Device Table on one line toward the bottom of the screen. This addition is useful to determine if some device might be used or hogged by another partition.

Function: Disk status
Modules: TBO.XDAD
Abstract: This program displays the status of all disks on the system. It is a static display which goes through possible disk addresses on 10, 20, and 30.

It shows Cur.end -Max., Access errors, disk not configured, disk is unavailable.

THIS IS THE STATUS OF POSSIBLE DISKS ON THE SYSTEM
1 310 Yes Cur.end=17414 Max.=38911 ! 2 B10.. etc.
3 350 Access error = 98
5

Function: CRT character set and keyboard values
Modules: TBO.CRT
Abstract: This program displays the full CRT character set. After the initial display, various keys can be depressed to determine the hex and ASCII values of keystrokes. (S.F.=hh) shows special function keys, i.e those with an ENDI level.
**Function:** T.C. Board status

**Modules:** TBO.STAT

**Abstract:** This program displays the status of peripherals on selected device addresses. Printers are tested on 215, 216, 217, and 218. T.C. boards are tested on 6 addresses from 1A to 1F. For T.C. boards the program shows Ready/Not, B,C, or D, and memory available. The status of the operating system and user terminal configuration are also displayed.

**BASIC-2 command** LIST DT shows addresses configured, selected or hogged.

---

**Function:** Examine ASKACALL file

**Modules:** TBO.XASK

**Abstract:** This program displays the contents of the configuration file "ASKACALL" used in the ASC, BSC, and 2200/3270 emulations.

- N$ is the name given the emulation by the operator,
- W$ in non-3270 displays the modules loaded.
- Z$ bytes 01-20 in hex — are parameters for the microcode.
- bytes 21-64 in ASCII are responses to prompts.
Tool Box Option Utility

III. Operating Instructions for selected utilities

Sort the disk catalog area  2 pages
Clock and calendar  2 pages
Disk status
T.C board, Printer and Terminal status  1 page.
LOAD RUN xx  1 page.
Cross-reference listing  3 pages
Program or system comparison  2 pages
Analyze data file structure  1 page
File Z A P  2 pages
Abstract: Sorts the entries of the disk catalog into an ordered sequence. The listing may be in name sequence or in sector number, or reverse sector number order. Program or data file names are listed item by item in separate groups. Listings requested by name may include all or a subset of the files stored.

Modules: TBO.SDC0 - TBO.SDC1

Equipment used: CRT / keyboard and optional printer.

Operating instructions:

Display REQUEST NUMBER= 1 SORT DISK FILE CATALOG NAMES
1 =310 2 =B10 3 =320 4 =B20

Prompt 1 Disk surface Key single digit or any valid disk address 1
Respond EXEC
  or digit and EXEC
  or disk address and EXEC
Display 1 =310

Prompt 2 Sort by 0=Name 1=Sector 2= S =Sector (DEFAULT)=O ? _
Respond EXEC for sequence by Name
  or 0 and EXEC for sequence by Name
  or 1 and EXEC for sequence in Sector number order
  or 2 and EXEC for sequence in -Sector order, ie. last names entered

Prompt 3 Common Root ID ? _ Asked if Prompt 2 response was 0.
Respond EXEC to collect names of all files
  1-8 characters and EXEC to collect by common ID. A virgule ("/"), in
  any position may be used for masked searching.

Display TYPE OF FILE NAMES TO SORT
PROGS+DATA A S ALL
PROGRAMS AP SP P=AP+SP
DATA AD SD D=AD+SD

Prompt 4 Category Active Scratched (DEFAULT)=AP? _
Respond EXEC for sequenced list of Active Program file names.
  or AP and EXEC for Active Programs.
  or SP and EXEC for Scratched Programs.
  or AD and EXEC for Active Data file names.
  or SD and EXEC for Scratched Data file names.
  or A and EXEC for Active Program and Data file names.
  or S and EXEC for Scratched Programs and Data file names.
  or ALL and EXEC for sequenced list of AP, AD, SP, and SD file names.
  or DATA and EXEC for sequenced list of AP, AD, SP, and SD file names

Prompt 5 Output to: 0=CRT 1=215 2=204 3=216 (DEFAULT)= 0 ? _
Respond EXEC or digit and EXEC.

Prompt 6 List wanted 0=Cat data 1=plus REMS (DEFAULT)= 1 ? _
Note "plus REMS" means to display a catalog index line and:
  for programs a portion of the first line of a program file if
  coded as a REM or % (image statement).
  for data files a description for the data file name if found in
  a list of DATA statements overlaid over lines 9000+.
Respond EXEC or digit and EXEC.
Prompt 7 INPUT NON-ZERO FOR MORE DATA? 
Respond EXEC to start Sort collection
or
key and EXEC to prompt for an additional sort collection.

Additional options:
These options may be invoked after the collection phase of the sort.

'1-Cat.only --- displays for each file only the single line of catalog information.

'2-Cat.with REMs --- displays for each file the catalog line plus a REM. (see prompt 6 elaboration).

'7-BEGIN --- starts the display over again with the first item.

'14 date + --- prompts for a date mm/dd/yy and then searches for program files entered that date or after. The program must have an initial REM or % statement with a date included. The included date may be of mm/dd/yy or yy/mm/dd format.

'15 RECALL ID
Skip through the list of collected names and begin with a matching ID.

Output:

Screen DISK CATALOGUE SORTED BY NAME
INDEX SECTORS = nn
END CAT. AREA = aaaaa
CURRENT END = aaaaa
SEARCHING FOR AP [Standard index/New index method
CAT. SECTOR= s FOUND ITEM = nn filename
AP ITEMS FOUND= ccc
ITEM NAME TYPE START END USED FREE +USED
1 TBO DIAG P S.s.# E.s.#  6 1   6
text of REM or % statement if first line of program.
2 TBO UTIL P S.s.# E.s.#  6 2 12
text of REM or % statement if first line of program.

'1-Cat.only  '2-Cat.with REMs  '7-BEGIN  '14 date +  '15 RECALL ID

Printer DISK CATALOGUE SORTED BY NAME
INDEX SECTORS = nn
END CAT. AREA = aaaaa
CURRENT END = aaaaa

AP ITEMS FOUND= ccc
ITEM NAME TYPE START END USED FREE +USED
1 TBO DIAG P S.s.# E.s.#  6 1   6
text of REM or % statement if first line of program.
2 TBO UTIL P S.s.# E.s.#  6 2 12
text of REM or % statement if first line of program.
Abstract: This program displays a clock and calendar on a 2200 terminal. The clock and calendar can be used for reminder messages for the day. If a universal global area is set reminders may be sent from terminal to terminal.

Modules: @CLOCK -- DATETIME -- TBO.MSGS

Equipment used: CRT / keyboard and MXE controller.

Operating instructions:

for "DATETIME" Display. Enter Date and Time MM/DD/YY HH:MM

Respond Valid date and time for the MXE controller.

for "@CLOCK" Display

FRIDAY JANUARY 24, 1986

WANG 2200 MICRO-V P

*** ******* ******* ******* ******* ******* *******
** ** ** ** ** ** ** ** ** ** ** ** ** **
** ** ** ** ** ** ** ** ** ** ** ** ** **
** ** ** ** ** ** ** ** ** ** ** ** ** **
** ** ** ** ** ** ** ** ** ** ** ** ** **
** ** ** ** ** ** ** ** ** ** ** ** ** **
****** ******* ******* ******* ******* *******

JANUARY 1986

SUN MON TUE WED THU FRI SAT
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30

FEBRUARY 1986

SUN MON TUE WED THU FRI SAT
1
2 3 4 5 6 7 8
9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28

Respond space to find instructions

Key NOTE (s.f. '6') to display messages for this terminal
-- permanent reminders are DATA statements at lines 6100+
Key INSERT to enter current reminder messages
Key DELETE to remove messages
Key LEFT or RIGHT to shift months display
Key CANCEL/EDIT to return to main menu
Key EXEC to show current month

Respond CANCEL/EDIT to exit to main module saving outstanding reminders.

NOTE: Reminder messages to display in BOLD TYPE and standard type may be inserted to occur at set times. Nine characters are displayed BOLD TYPE.

NOTE: Holidays and weekends are displayed in highlighted reverse intensity. The current day is displayed in boxed in high-intensity blinking.

Holidays are coded in A$="mm/dd, ... " form on line 125.
Additional instructions for @CLOCK.

hh:mm START.JOB FileName — Special NOTE form to LOAD RUN "FileName"
Key '31 to send a message to another terminal.

respond '10 to INSERT messages to your terminal's Clock file.
display

hh:mm BOLD TYPE insert message
hh:mm text for message 1
hh:mm text for message 2
respond hh:mm text for new message
or
mm/dd hh:mm text for new message

respond '9 to DELETE messages from your terminal's Clock file.
display

hh:mm or ALL — DELETE message
hh:mm text for message 1
hh:mm text for message 2
respond hh:mm and RETURN to delete one of today's messages
or
mm/dd hh:mm and RETURN.
or
ALL and RETURN

Enhancements for an improved clock functionality.

1. File "SYS CRTS" may be added to describe terminals on your system.
   NOTE descriptions for system terminals are in a file "SYS CRTS"

   8000 REM .SYS CRTS
   :REM. "Msg.File # CRT description
   8010 DATA "@CLOCKM" 1 Main system terminal
   8020 ... "MSGS.TBO 2 Main system terminal

2. A global message area may be established for inter-terminal communications.
   Line 24 of @CLOCK looks for this partition.
   0024 MD=1
   : SELECT @PART "3270UNIV"
   : ERROR MD=0

3. Use a file called "SYS COM" as the first Universal COMmon memory module.

   0010 REM SYS COM mm/dd/yy COMmon memory for this system
   0015 REM .Called from WAITDATE
   0020 COM @M$(8)50
   : REM /* Terminals for CLOCK messages.
   0070 LOAD T "3270U2" Calls next Universal partition module.
**Function:** What disks are on the system?

**Modules:** TBO.XDAD

**Abstract:** This program displays the status of all disks on the system. It is a static display which goes through possible disk addresses on 10, 20, and 30.

It shows Cur.end, Max., Access errors, disk not configured, disk unavailable.

What disks are attached to the system

<table>
<thead>
<tr>
<th>Disk controller</th>
<th>Disk type</th>
<th>1 310 Yes Cur.END=17414 Max.=38911</th>
<th>2 B10 ERR.I98 Platter not mounted</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 310 Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2 D11 Yes</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3 D12 ERR.I91</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Disk controller 20 Disk type CO

| 1 320 Yes Cur.END= 1231 Max.= 1231 | 2 B20 Access error = 98 |
| 3 360 ERR.I98 Platter not mounted |

| 1 330 Yes Cur.END=17650 Max.=65023 | 2 B30 etc. |
| 3 370 Yes ..etc                   |
| 5 D31 Yes ..etc                   |

**THIS IS THE STATUS OF POSSIBLE DISK ADDRESSING ON THIS SYSTEM**

Key '0 to EXIT '1 to TBO.STAT '15 RECALL - other for disk notes

(canned informational display)

| Shugart -- Dual or triple (1231) | maximum sectors, white label |
| Winchester (3873) maximum sectors, red label |
| 2275's -- DSDD max sectors (1292) maximum sectors, 5 1/4 inch |
| 2275-30 -- DSDD on D.0 (1292) Winchester on D.1 and D.2 (18900) |
| 2275-60 -- No diskette -- 4 Winchester surfaces on D.0 thru D.3 (65023) |
| Phoenix -- No diskette -- surfaces on D.0 thru D.F (52763) |
**Function:** Show status of Printers, T.C. boards, and Terminal

**Abstract:** This program displays the status of peripherals on selected device addresses. Printers are tested on 215, 216, 217, and 218. T.C. boards are tested on 6 addresses from 1A to 1F. The status of the operating system and user terminal are also displayed.

**Modules:** TBO.STAT

Screen display

2200 Device status

---

Terminal status

`!*2436DE R0101 19200B 8+0 (USA)!`

<table>
<thead>
<tr>
<th>Terminal number</th>
<th>1</th>
<th>CPU type</th>
<th>MVP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Partition number</td>
<td>9</td>
<td>MVP Release</td>
<td>2.6</td>
</tr>
<tr>
<td>Partition memory</td>
<td>56</td>
<td>CRT size</td>
<td>80</td>
</tr>
</tbody>
</table>

**Printers**

- 215 Unavailable
- 217 Unavailable

**T.C. Boards**

- 01A Unavailable
- 01C B. 32K
- 01E B. 8K

**Key 'O to START '1 to What disks?** other key to retest here
Function: LOAD RUN xx from another surface

Abstract:

The program does a SELECT #1 with the given disk address. It then tries a LIMITS T #1, with the file name given to assure the file name given is a valid program on the stated disk surface.

If false, reprompting occurs. S.F. '0 may be used to reload "START".

If true, TBO.LRUN does a SELECT #0 with the new disk surface; the new program is activated with a LOAD RUN command.

Modules: TBO.LRUN

Display Activate a system from another platter

Prompt 1 On disk surface ___
Respond  hhh and EXEC  where hhh is a valid disk address.

Prompt 2 Program START___
Respond  EXEC to load "START".
else  file name and EXEC.
Function: Cross-reference listing

Abstract: One or more program files on any system disk may be listed on numbered and titled pages. For each program requested the listing consists of decompressed program statements followed by a cross-referenced listing of variables used, special function, and numbered statement references.

Modules: TBO.CRF0 - TBO.CRF1 - TBO.CRF2 - TBO.CRF3

Equipment used: CRT / keyboard, disk and optional printer.

Operating instructions:

Display TBO xref — System code = MVP with 56 K memory
   CROSSRF — S.F. Entry points
   From prompt module:
   '0 Change Disk media
   '1 Pick-up prompting in program names
   '5 display verb atoms   '15 Show S.F. actions
   '12 writeup       '14 List options
   From cross-reference listing mainline:
   '10 Summary to Printer/CONTINUE   '11 Summary only

Prompt 1 Output to:  0=CRT  1=215  2=204  3=216  (DEFAULT)= 0 ?
Respond EXEC or digit and EXEC.

Prompt 2 DATE?
Respond any character stream (no commas) will be part of a large print title put at the top of each printed page.
   EXEC or digit and EXEC.

Prompt 3 1=SAVE MEANINGS THRU ALL PROGRAMS?
Respond EXEC if you are not sure.
   or 1 and EXEC if variable meanings are described and are to be carried from program to program.

Prompt 4 For annotation KEY — 0=Summary only 1=In Listing 2=In Margin
   DEFAULT=0 ?
Respond EXEC if you are not sure.
   or 1 and EXEC to get meanings embedded in the listing.
   or 2 and EXEC to get meanings placed in the right margin of the listing.

Prompt 5 PAPER WIDTH NARROW=0 WIDE=1 DEFAULT=0 ?
Respond EXEC or 0 and EXEC for 80 column listings
   or 1 and EXEC for 120 column listings

Prompt 6 List mode   KEY--0=List + XRef 1=List only  2=XRef only
   DEFAULT=0 ?
Respond EXEC or 0 and EXEC to get a complete listing.
   or 1 and EXEC to get a listing of only the program text,
   or 2 and EXEC to get a listing of only the cross-reference summary.

Prompt 7 START STMT DEFAULT=0 ?
Respond EXEC to get a listing of all lines of the program.
   or 1 and dddd EXEC to get a listing beginning at a specified line.

Prompts 8 and 9 are repetitive through a list of file names.
Prompts 8 and 9 are repetitive through a list of file names.

Display 1 = 310 2 = 810 3 = 320 4 = 820
Prompt 8 Disk surface Key single digit or any valid disk address 1
Respond EXEC
or digit and EXEC
or disk address and EXEC
Display 1 = 310

Prompt 9 Disk Program Name + commentary. S.F.'1 ON ERROR 80)?
Respond Program name (8 characters) plus commentary to append to title.
and EXEC prompt 9 will repeat until a single EXEC is entered.
or DATA and EXEC to get a list of program names from a list of DATA
statements which may loaded to overlay the code at 9000+.
or ALL and EXEC to localize the search by a common 0-8 byte IDentity.
or ALL- and EXEC to exclude modules with a common 1-8 byte IDentity.
or '0' to select an alternative disk surface.
or EXEC to terminate the list of items.

NOTE: once the prompting sequence has gotten to Prompt 8 or Prompt 9 the
S.F. keys are very useful to pick-up within the sequences.
S.F. '0' may be used to select a new disk surface,
and S.F. '1' to pick-up in the list of file names.

Output:

Section 1 - An index page to the program files listed, disk surface and
description.

Program listing:
Titled and page numbered listings of program files.
Section of decompressed program code.

Summary cross-reference section
Variables

Special functions
Statement numbers.

The cross-reference program will create a listing of any BASIC
program file. The listing will consist of titled and numbered pages. Each
listing contains a program listing where each segment of a multi-statement
line appears on a separate printer line. The listing will contain a blank
line after each GOTO, RETURN, LOAD, or STOP statement. The listing will be
automatically indented following IF statements, if not followed by another IF
statement, and following FOR statements.

A separate summary section lists:
1) # references and variables used, meanings (optional), and
   referencing statement lines.
2) Special functions and meanings (optional).
3) Statement number and special function cross-references.
SPECIAL OPTION — PAGE FORMATTING:
The CROSSRF program is written to interpret certain REM statements in
a way that will enhance program documentation. REM statements may be
incorporated into the source BASIC causing special listing effects.

REM% - REM (per cent) (up arrow) (comment)
positions to a new printer page with title and expanded print comment.

REM% - REM (per cent) (digit 1-9) (comment)
skips n lines and prints the comment in expanded type.

REM% comment - REM (per cent) (comment)
skips 2 lines and prints the comment in expanded type.

:REM/ comment - (colon) REM (slash) (comment)
puts the comment in the right column of the previous statement.

SPECIAL OPTION — ENTRY OF MEANINGS TO VARIABLES AND SPECIAL FUNCTIONS:

A descriptive meaning may be assigned to each variable, # file
reference, or special function. This meaning, if used will be automatically
output in the cross-reference summary listing. The meaning may also be output
in the body of the listing either embedded into the program code or in the
right margin on each occurrence of the variable or special function. The
descriptive material is entered into special tables by special REM
statements. The special statements, described in the below, may be entered
either in the program text or in a separate program file. Any variable may be
given a new meaning merely by entering a new meaning item. Refer to question
3 in the interactive sequence for activating the meanings option.

REM%0 -REM (per cent) (zero) causes an entry to be placed into
the meanings table. Meanings entries are coded as follows:

name space description comma
or
name space description colon
or
name space description carriage return
where name is up to 4 characters with no spaces, eg AO$(
space is one or more spaces,
description is 1 to 16 characters of description (no commas)
comma, colon, or carriage return terminates the meaning.

Example of BASIC program code required to enter meanings.

30 REM%0 A A, B CRT line, C Start char pos, C$ C$1, L Field size
40 REM%0 '201 Edit input, #5 CRT

TABLE CAPACITIES
The CROSSRF program does not use any storage other than CPU memory.
In order to create cross-reference summaries it is necessary to store certain
table in system memory. Memory space for these tables is coded into COM
statements in the loader module (TBO.CRF0). A separate set of COM statements
for each memory size has been incorporated on the program disk. The default
values for COM statements of various memory configurations may be modified if
necessary.
Function: Program or system comparison
Modules: TBO.CMPS
Equipment used: CRT / keyboard, disk and optional printer.

Abstract: Program files on any system disks are compared line by line. Differences between the two files are displayed line by line. Two system disks may be compared by activating a list of DATA statement names.

The program comparison utility operates in one of two modes:
1). **program mode** allows the comparison of single program files. The programs may have the same name on different disk surfaces or different names on the same disk surface.
2). **system mode** allows the comparison of a list of program files occurring on two different disk surfaces. The list of names is from DATA statements overlaid over lines 9000 and beyond. The operator keys DATA when prompted for the first input file name.

Output to the CRT is scrolled and will show:

```
#1 IS ON LINE = 0010
#2 IS ON LINE = 0010

#1 IS ON LINE = 0020
#2 IS ON LINE = 0020

#1 --
  0020  COM @Z$(26)80, @X$(2)2, ... text for line with differences.

#2 --
  0020  COM @Z$(26)80, @X$(1)2, ... text for line with differences.
```

If output is to the CRT only, the program will pause at each differing line. The operator may key any key to continue the search. Keying RUN will allow the program to continue the search of the current files without pause.

If output is to the printer the program will continue without pause through the entire list of programs to be compared.
Operating instructions:

Display 

PROGRAM COMPARISON
!

NOTE— TO COMPARE SYSTEMS WITH INPUT VIA DATA STATEMENTS
ENTER THE NAME DATA AS THE 1ST PROGRAM NAME FOR #1

Key S.F. '1' to pickup in entry of program names

prompt KEY 0 if output to CRT?
respond 0 and RETURN for output to the CRT
or 1 RETURN for output to Printer

additional prompts for printer output only

prompt # OF COPIES?
prompt COMMENT?
respond comments for up to 9 lines of commentary for the top of the listing.
or space and RETURN to end the COMMENTS.

display FIRST PROGRAM—#1
1 =310 2 =B10 3 =320 4 =B20

Prompt Disk surface Key single digit or any valid disk address 1__
Respond EXEC
or digit and EXEC
or disk address and EXEC
Display 1 =310
prompt PROGRAM #1 NAME? __________
respond 8 character program name and EXEC
or DATA and EXEC

display SECOND PROGRAM—#2
1 =310 2 =B10 3 =320 4 =B20

Prompt Disk surface Key single digit or any valid disk address 1__
Respond EXEC
or digit and EXEC
or disk address and EXEC
Display 1 =310

the prompt below will occur if DATA was not entered for PROGRAM #1.
prompt PROGRAM #2 NAME? __________
respond 8 character program name and EXEC
or DATA and EXEC
Function: Analyze data file structure

Abstract: This program analyzes a catalogued data file for structure. The output is a summary of the SAVE statements used to create the file.

Example

File name = @SYSFILE
File size = 32 Sectors.
File = @SYSFILE BASIC-2 data structure -- SAVE number 1
$8 Scalar $32 $(2)16 $(16)8 $(33)3 $(15)13

Example

File name = BSC*010A
File size = 40 Sectors.
File = BSC*010A BASIC-2 data structure -- SAVE number 1
$(256)16
File = BSC*010A BASIC-2 data structure -- SAVE number 2
$(256)16
END OF FILE

Operating instructions:

Display Analyze Data File Format

$4)62 Wang T.C. file format
$2, $3, $(4)60 Wang Prom file format
$(256)16 4K TC microcode
$(512)16 8K TC microcode
$(34)121 4K TC ucode (min.disk storage)
$(3)83 3270 Spooler names file
$(3)80 3270*PQ
$31, $(128)2, Scalar(3) VFU format tape
$8, Scalar, $32, $2(16), $(16)8, $(33)2 @SYSFILE

Key '15 to see file examples
1 =310 2 =B10 3 =320 4 =B20

Prompt Disk surface Key single digit or any valid disk address 1__
Respond EXEC
or digit and EXEC
or disk address and EXEC
Display 1 =310

display see example above.
**Function:** File Z A P  

**Modules:** TBO.ZAP  

**Abstract:** A single disk surface may be examined. The display shows the contents of a single sector in hexadecimal notation and in ASCII.

1. Items in the catalog index area are flagged as AP (Active Programs), AData (Active Data).
2. Data files are displayed with highlighted attribute bytes.
3. Program files are displayed with highlighted RETURN codes.

Caution, you have the ability to change any all data in a selected sector.

The CRT will show a full screen non-scrolling display of the form below. The contents of the boxes pictured as (alpha) are described in the writeup below.

<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)</td>
<td>(B)</td>
<td>(C)</td>
<td>(D)</td>
<td>(E)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(F)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Wang Laboratories, Inc.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>SUPERZAP Ver. d.d.d.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(G)</td>
<td>(H)</td>
<td>(I)</td>
<td>(J)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(K)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **(A)** Sector = ddddd contains the disk sector number being examined.
- **(B)** Device = hhh contains the disk surface being examined.
- **(C)** (alpha text) Shows a MODE: eg. blank, Loading, printing, display
- **(D)** (alpha text) Prompt entry field.
- **(E)** (alpha text) Prompt entry field.
- **(G)** (alpha text) In the catalog shows AP and AData items.
    - In a named catalogued data file shows variable sizes.
    - In a named program file shows the initial line number.
- **(H)** (hex values) Values in hex of the sector being examined.
    - In a named catalogued data file highlighted SOW bytes.
    - In a named program file highlighted Carr. Return bytes.
- **(I)** (ASCII text) Values in ASCII of the sector being examined.
- **(J)** Description of possible operations.
- **(K)**

<table>
<thead>
<tr>
<th>'0'</th>
<th>Change sector</th>
<th>'9'</th>
<th>Find ASCII string</th>
<th>'14'</th>
<th>Print sector</th>
</tr>
</thead>
<tbody>
<tr>
<td>'1'</td>
<td>Change device</td>
<td>'10'</td>
<td>Find HEX string</td>
<td>'11'</td>
<td>Find start of file</td>
</tr>
<tr>
<td>'3'</td>
<td>HEX to decimal</td>
<td>'12'</td>
<td>Next sector</td>
<td></td>
<td></td>
</tr>
<tr>
<td>'4'</td>
<td>Exit program</td>
<td>'13'</td>
<td>Previous sector</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Prompts for File ZAP

Prompt 0. Change sector
respond 'ddddd and RETURN valid disk sector address.

Prompt 1. Change device
respond 'hhhh and RETURN valid 3-hexdigit disk address.

Prompt 3. HEX to decimal
respond 'hh and RETURN valid hex values.

Prompt 4. Exit program

Prompt 9. Find ASCII string
respond text and RETURN valid ASCII data to find.

Prompt 10. Find HEX string
respond text and RETURN valid HEX data to find.

Prompt 11. Find start of file
respond valid file name and RETURN.

Prompt 12. Next sector
Prompt 13. Previous sector
Prompt 14. Print sector
Abstract: From the "START" module this function analyses a disk surface to find and correlate all program and data file references. Program modules found are of two types: menu modules are a list of DATA statements originated at BASIC-2 line numbers 9000 and above. Function modules are all other program modules and are either from menus or other programs. The program operates in two parts. Part one "TBQ.DMPM" goes through the menu structure and creates two COMMON tables listing menu modules and non-menu program modules. Part two "TBQ.FMPN" goes through all the program function modules to find all program LOAD and LOAD at DATA LOAD OPEN statements.

Equipment: Equipment used: CRT / keyboard and optional printer.

Operating Instructions:

Display
Function Analyzer
Source Program disk
1 = T10
2 = T10
3 = T320
4 = T20
Prompt 1 Disk surface - Key single digit or any valid disk address
Respond EXEC
or EXEC
or disk address and EXEC
Display 1 = T10
List of programs is from COMMON.PS() array
1 START
2 @CLOCK

Program Name?
Respond EXEC
or program name and EXEC
KEY = EXEC.FOR HARD COPY?

Display
(High-lighted Program name) = ..SELECTED LINES Prog. 1 of 16
(listing of all LOAD statements found in program)

(High-lighted Program name) = VERB = LOAD IS USED d TIMES
Referenced:
PROGRAM LOAD is at 0010
LOAD at 0010

Example
Example
STRT3275
..SELECTED LINES Prog. 28 of 16

0040 DATA LOAD DC OPEN T #0, "M-3270"
0660 DATA LOAD DC #0, W$()
0150 LOAD DC LOAD DC T "EM3275"

STRT3275
STRT3275
VERB=LOAD IS USED 3 TIMES

Files referenced

D M-3270
F Function EM-3275
1/21/94  If select SCSI pick from BACKUP, RESTORE, or the DISK MANAGEMENT window, returns you back to the menu with a message.

Fix: added line 25 to all 3 programs

25 IF STR(A$9,1) = "T" THEN 40: PRINT AT (3,0); HEX(0E); "Device Not Supported!!"; HEX(0F 01); REM - NON-TURBO CPU
B. TAGG (IF THEN ELSE TEST)

GBT ERROR EVERY PASS & NEVER LEAVES THIS SCREEN OF TESTS.

8/18/94 ELSE STATEMENT IGNORED ON CS/2200 O/S 2.7 & BELOW IF NOT ON SAME LINE AS IF/THEN.

LINE 730 INCORRECTLY TESTS X <> Ø. SHOULD BE TESTING X <> 2 AS IF WORKING CORRECTLY X WILL ALWAYS BE EQUAL TO 2.

FIX

700 X = Ø: IF 1 < 2 THEN X = 1: ELSE X = 2
710 IF X <> 1 THEN 8990
720 X = Ø: IF 2 < 1 THEN X = 1: ELSE X = 2
730 IF X <> 2 THEN 8990
CHANGES for MA TRUCK BODY in Chelsea, MA for year 2000

MB1NKFAM - used to set system date on boot. Made copy - MBN1KKFA1

MB1NKFAM line 3080 deleted: IF U9 = Ø THEN 3100 AT END

TEM1B/UP - customized copy of CB1ACKUP for TRAILER EQUIPMENT SALES

TEM1NB/UP line 900 deleted: IF U9 = Ø THEN 920 AT END

900 REM CHECK FORM: IF VER(U98, "###/###/###") <> 8 THEN 920:
REM CHECK MO.: CONVERT STR(U98,1,2) TO U9: IF U9 < 1 OR U9 > 12 THEN 920:
REM CHECK YR.: CONVERT STR(U98,7,2) TO U9

CB1ACKUP-5/13/98 VER. deleted: IF U9 = Ø THEN 920 AT END OR LINE 900

TO ALLOW USE OF YEAR ØØ
CHANGES for MA TRUCK BODY in Chelsea, MA for year 2000

MBNKKFAM = used to set system date on boot. MADS COPY = MBNKKFAM

MBNKKFAM Line 3080 deleted: IF U9 = Ø THEN 3100” at end

TEMNB/UP = CUSTOMIZD COPY OF "BACKUP FOR TRAILER EQUIPMENT SALES"

TEMNB/UP Line 900 deleted: IF U9 = Ø THEN 920” at end

NEW 900 REM CHECK FORM: IF VER.(U98, "##/##/##") <= 8 THEN 920:
REM CHECK MO.: CONVERT STR(U98, 1, 2) TO U9: IF U9 < 1 OR U9 > 12
THEN 920: REM CHECK YR.: CONVERT STR(U98, 7, 2) TO U9

BACKUP-5/8/98 ver. deleted: IF U9 = Ø THEN 920” at end of Line 900

to allow use of year ØØ

INSTALLATION INSTRUCTIONS:

AFTER SYSTEM IS UP & TERMINAL 1 IS AT MAIN MENU # 1:

1. INSERT FLOPPY DISK IN FLOPPY DRIVE

2. SELECT ‘16’-DISK UTILITIES FROM MAIN MENU

3. USE SPACE BAR TO SELECT ‘MOVE FILE’ & KEY RUN

4. CHANGE "INPUT ADDRESS" TO DIO TO READ FROM FLOPPY

5. KEY RETURN TWICE TO ACCEPT DIO ADDR & WANG PLATTER TYPE

6. CHANGE "OUTPUT ADDRESS" TO DII

7. KEY RETURN TWICE TO ACCEPT DII ADDR & WANG PLATTER TYPE

8. ENTER Y WHEN ASKS IF WISH TO MOVE ALL ACTIVE FILES

9. KEY RETURN TWICE TO ACCEPT CHANGE & TO OVERWRITE FILES.

10. 3 FILES WILL BE UPDATED. KEY CANCEL/EDIT TO RETURN TO MENU WHEN DONE
CHANGES FOR MA TRUCK BODY IN CHELSEA, MA TO ADD ACCESS TO DISK UTILITIES MENU FROM THEIR MAIN MENU + TO INCREASE PARTITION SIZE TO ALLOW DSTAPEB, TAPE BACKUP, TO RUN.

From DISK UTILITIES 1.10.00, THE FOLLOWING FILES WERE TAKEN WITHOUT CHANGE:

- STARTD
- CDSAPPLY
- CDS APPLY
- CDS CONFIG
- CDS CONFIG
- CDS TAPER
- CD TAPER
- CHITRATE
- CTO IMAGE
- CTO CREF
- CTO CRE6
- CTO DISK
- CTO SUBS

From DISK UTILITIES 1.10.00, THE FOLLOWING FILES WERE TAKEN + EDITED AS SHOWN:

1. START LINE 10 MADE #PSTAT= " " INSTEAD OF "SYSMVPA"
   RENAMED PROGRAM TO DISKUTIL.

   CHANGE TO #PSTAT ALLOWS DEFAULT TO ROOT NODE + CAN PREVENT PROBLEMS CAUSED BY COMMON VARIABLES. HAD TO RENAME PROGRAM AS START ALREADY EXISTED.

2. CMENU LINE 120 AFTER 1ST : ADDED IF UB="TEMNMENU" OR UB="YES"
   THEN 125:
   ADD NEW LINE 125 LOAD RUN #U2, "TEMNMENU"

   CUSTOMER NOT USING CMENU FOR THEIR MAIN MENU. WHEN KEYING CANCEL/EDIT TO RETURN TO CUSTOMER'S MENU, TEMNMENU, WOULD GET PS6 ON LINE 720. TEMNMENU WAS OVERLAYING CMENU CAUSING A MESS OF BOTH PROGRAMS.

3. CMOVEFIL LINE 550 CHANGED LOAD RUN "START" TO LOAD RUN "DISKUTIL"
   CHANGE REFLECTS RENAMING OF START TO DISKUTIL. LINE IS USED WHEN KEY FN TAB TO RETURN TO DISK UTILITIES MENU.

4. CSYSMVPA LINE 9200 CHANGED "SYSMVPA" AT END TO "TEMNMENU"
This IS THE DISK UTILITIES MENU + THIS CHANGE ALLOWS YOU TO CANCEL/EDIT BACK TO THE CUSTOMER'S MENU, TEMNMENU, INSTEAD OF LOOKING FOR SYSMVPA.
5. C.BACKUP  LINE 9040 REM'd LINE
   ELIMINATES SCSI TAPE BACKUP FROM BACKUP UTILITIES MENU.

6. C.DISK    LINE 9050 REM'd LINE
   ELIMINATES SCSI CONFIGURATION FROM DISK MANAGEMENT MENU.

7. C.RESTOR  LINE 9040 REM'd LINE
   ELIMINATES RECOVERY FROM SCSI TAPES FROM RESTORE UTILITIES MENU.

The following customer file was edited:

TEMNMENU  added new line 4115 PRINT TAB(43);"'16 - DISK UTILITIES"
line 4150 added 4325 before :GOTO 4130 at end
added new line 4325 GOSUB '33("DISK UTILITIES");COM CLEAR;
load DCT #0,"DISKUTIL"

Adds DISK UTILITIES pick to customer's main menu.

The DS tape backup program, CDSTAPEB, would not run in a 23.5K
partition the customer was using. Created new configuration file,
MATRUCK, in CGENPART with larger partition sizes to take
advantages of added memory. Old configuration file, "TESI", still
exists with customer's original configuration. New configuration:

Partition  Memory  Terminal  Programmable Program to Load
  1  u  3.0  1  y  KFAM/417
  2  47.0  1  y  TESI
  3  11.0  1  y  MBNKKFAM
  4  45.0  2  y  TEMNMENU
  5  11.0  2  y  MBNKKFAM
  6  56.0  3  y

Related files included: CGENPART (latest version), CSYSFILS (configuration file)
To install changes to add backup utilities to system:

1. Make sure no one is using system and all jobs are completed.
2. From Main Menu #1, key Shift/Reset. Screen is blank except for "READY (BASIC-2) PARTITION OZ" at top.
3. Type in CLEAR then Hit Return key.
4. Type in SELECT DISK D10 then Hit RETURN key.
5. Insert diskette with changes in floppy drive & close latch.
6. Type in LOAD RUN "DISKUTIL" then Hit RETURN key.

Disk Utilities Main menu should display on screen.

7. Use space bar key to move attention block to "MOVE FILE" pick.
   Hit Run key to run Move File program.
8. Screen should now be prompting you for "INPUT ADDRESS: DI1"
   Change this address to D10 then key RETURN.

   If at any of the fields in the move file utility you make a mistake & need to go back to correct, hit the fn or Tab key & restart at Step 7.
9. Now asks for "INPUT PLATTER TYPE: W". Hit RETURN to leave as W.
10. Change Output Address to D11 & Hit Return key.
11. Hit Return to leave "Output Platter" as Type W.
12. Make sure Input Address = D10, Output Address = D11 & both are Types WANG. Key FN/Tab key & restart at Step 7 to correct.

System is now asking if you wish to move all files. Change the N to Y & key Return.

13. Now asks if you want to overwrite. Leave as Y & key Return.

System will now begin moving files to your system disk. When done (about 5 min), system will return to Disk Utilities screen.
14. System must now be rebooted to utilize changes made. Key SHIFT/RESET to clear screen.

15. Type in BINIT"SYSTEM" then RETURN to reboot. SHOULd come back w/ MOUNT SYSTEM PLATTER PRESS RESET

16. Reboot system as you normally would from this screen.

At your MAIN MENU #1 you will now see an additional pick, '16 - DISK UTILITIES
Use $F 16 (SHIFT/SF 0) to access the DISK UTILITIES from which you can run your BACKUP.

To run BACKUP from MAIN MENU #1,
2. Use space bar to move alternating block to BACKUP UTILITIES & KEY RUN.
3. Backup disk platters to DT Tape Cassette should be selected. Key RUN.
4. Screen prompts for tape address. $5F is correct. Hit RETURN.
5. Append or Erase. Append allows you to add this backup to current end of tape allowing previous data or backups to remain. Erase clears tape. Could probably fit 4 or 5 backups of DII on 1 tape before running out of space. Choose A or E & KEY RETURN.

6. Now asks for address of disk to backup. Type in DII & RETURN.
7. Enter label for backup minimally today's date & KEY RETURN.
8. START Sector must be 0. KEY RETURN.
9. Accept default end sector by KEYING RETURN.
10. KEY RETURN to accept entry. If need to correct hit N RETURN & go to 6
12. If DII is only backup blank out address & KEY RETURN, otherwise repeat steps 6 through 10 for each address to backup.
13. When backup completes, key FN/TAB key. Returns system to Backup Menu.
14. Key CANCEL/EDIT to return to Disk Utilities screen.
15. Key CANCEL/EDIT to return to Main Menu #1.