
Appendix C. A Sample Program for Filling Out the Planning Charts

The listing below will allow you to create a program that will assist you in filling out the planning documents described in Chapter 3, with the exception of the IBM 8230 Planning Chart. If you use this program to fill out the various planning documents initially, you will be able to record changes to your network with a minimum of paperwork, since the program provides you with a permanent electronic record of your data.

If you have already used the listing in the edition of this manual numbered GA27-3677-1, the listing which follows the main program listing can be used to create a program to convert your existing data files to the new format required by this program. Before using this program to convert your data, be sure you have a backup copy of your data available.

This sample program has been designed for use with the IBM Personal Computer (except the IBM *Portable* Personal Computer) or its equivalent. It may be run under DOS 2.0 or higher and requires Advanced BASIC.

To use this program, do the following: Type the listing into your computer using your editing program, being careful to enter it without mistakes. (Each program statement must be entered on a single line. Long statements are printed in broken form only because of the size limitations of the page.) After you have entered the entire listing, save the file on a formatted diskette, using LAN4.TXT as the filename.

To run the program, first type in the following line at the DOS prompt and press Enter:

```
BASICA/S:200
```

The "/S:200" increases the maximum file buffer size for random files to 200 bytes. This is necessary because the record length of several of the chart files exceeds the default maximum size of 128 bytes.

Next, load the program by entering:

```
LOAD"LAN4
```

When the program has been loaded, BASIC will respond with an "OK" prompt. To start the program, enter:

```
RUN
```

The menu appears on the display. From the menu, you may select an option by entering a number. You may exit the program by pressing the Enter key alone. Exiting the program this way will return you to BASIC. To get back to DOS, enter the "SYSTEM" command.

The data input panel in this program requires that you fill in the fields in order, pressing the Enter key after each entry. You may not use the tab keys to move from one field to another. Once a field has been passed, you cannot space back to it to change it. Incorrect entries can be changed only after all fields on the panel have been completed. If you make a mistake, use the Enter key to move through the remaining fields until you reach the end of the panel. The charts generated by this program have a somewhat different format from those provided in Appendix B of the manual due to the limitations of some printers used with personal computers. If you use these charts for installation purposes, be sure that the installation personnel understand the difference in format.

Sample Program Listing for LAN4

```
10 KEY OFF
20 DIM F$(24):DIM VAR$(24):DIM FLNG%(24):DIM FLOC%(24)
30 DIM ROW$(24):DIM ROW%(24):DIM COL%(24):DIM CCNT%(24)
39 '
40 FILE1%=1:FILE2%=2:FILE3%=3:RINGLG%=3
50 B2$=SPACE$(2):B3$=SPACE$(3):B4$=SPACE$(4):B6$=SPACE$(6):B8$=SPACE$(8)
60 B10$=SPACE$(10):B12$=SPACE$(12):B14$=SPACE$(14):B16$=SPACE$(16)
70 B18$=SPACE$(18)
80 VBAR$=CHR$(179):HBAR$=CHR$(196)
90 H3$=STRING$(3,HBAR$):H4$=STRING$(4,HBAR$):H6$=STRING$(6,HBAR$)
100 H8$=STRING$(8,HBAR$):H10$=STRING$(10,HBAR$):H14$=STRING$(14,HBAR$)
110 H16$=STRING$(16,HBAR$)
120 UR$=CHR$(191):LL$=CHR$(192):LR$=CHR$(217):UL$=CHR$(218)
130 BT$=CHR$(193):TT$=CHR$(194):LT$=CHR$(180):RT$=CHR$(195):CR$=CHR$(197)
140 LL2$=CHR$(200):LR2$=CHR$(188):CR2$=CHR$(206):HBAR2$=CHR$(205)
150 TT3$=CHR$(210):CR3$=CHR$(216)
160 FF$=CHR$(12):BH$=CHR$(27)+CHR$(69):EH$=CHR$(27)+CHR$(70) ' Printer ctl chrs
199 '
200 DDATE$=LEFT$(DATE$,2)+"/"+MID$(DATE$,4,2)+"/"+RIGHT$(DATE$,2) ' Save date
210 DEF SEG=0:IF (PEEK(&H410) AND &H30)=&H30 THEN CLR%=0 ELSE CLR%=1 'Dsp type
220 DEF SEG
230 FG1%=7:BG1%=0:FG2%=0:BG2%=7:IF CLR%=1 THEN BG2%=3
240 ON ERROR GOTO 4500 ' Branch to 4500 if error
299 '
300 RTN%=0:NOFILE%=0:OPEN1%=0:HEXFOLD%=0:COLOR FG1%,BG1% 'Init flags & scrn color
399 '
400 GOSUB 1000 ' Display main menu & get selection
410 IF RTN%=1 THEN CLS:END ' Nothing selected - end pgm
420 GOSUB 2000 ' Set var values for this chart
430 GOSUB 4000 ' Open file for this chart type
435 IF DATATYPE%=5 AND OPTNO%=1 THEN GOSUB 1500 'Get ring data rate for RSC
440 IF NEWFILE%=1 AND OPTNO%<>1 THEN NOFILE%=1:GOTO 400 'No file for prt/list
450 ON OPTNO% GOSUB 10000,20000,30000 ' Execute selected option
460 IF OPEN1%=1 THEN CLOSE FILE1%:OPEN1%=0 ' Close the data file
470 GOTO 300 ' Return to main option screen
997 '
998 '***** DISPLAY MAIN OPTION SCREEN AND GET USER SELECTION NUMBER *****
999 '
1000 CLS
1010 LOCATE 1,23:PRINT "IBM TOKEN-RING NETWORK RECORD KEEPING"
1020 LOCATE 4,27:PRINT "1) IBM 8228 Cabling Charts"
1030 PRINT TAB(27) "2) IBM 8218 Cabling Charts"
1040 PRINT TAB(27) "3) IBM 8219 Cabling Charts"
1045 PRINT TAB(27) "4) IBM 8220 Cabling Charts"
1050 PRINT TAB(27) "5) Ring Sequence Chart"
1060 PRINT TAB(27) "6) Locator Charts"
1070 CROW%=CSRLIN+1:CCOL%=58
1080 LOCATE CROW%,29:PRINT "Select a CHART TYPE (1-6) =>"
1090 LOCATE CROW%,CCOL%:COLOR FG2%,BG2%:PRINT " ":COLOR FG1%,BG1%
1100 LOCATE CROW%+3,27:PRINT "1) Enter/change/delete/display data"
1110 PRINT TAB(27) "2) Print chart"
1120 PRINT TAB(27) "3) List all entries"
1130 TROW%=CSRLIN+1:TCOL%=52
1140 LOCATE TROW%,29:PRINT "Select a TASK (1-3) =>"
1150 LOCATE TROW%,TCOL%:COLOR FG2%,BG2%:PRINT " ":COLOR FG1%,BG1%
1160 IF NOFILE%=1 THEN GOSUB 1750
1170 LOCATE 24,27:PRINT "or press ENTER to end program";
```

```

1199 '
1200 R%=CROW%:C%=CCOL%:LVAL%=0:HVAL%=6:GOSUB 1800 ' Get chart type selection
1210 IF A%=0 THEN RTN%=1:RETURN ' Nothing selected -- return
1220 DATATYPE%=A%
1230 IF NOFILE%=1 THEN NOFILE%=0:RR%=21:GOSUB 1600 'Reset error flag & msg line
1240 R%=TROW%:C%=TCOL%:LVAL%=0:HVAL%=3:GOSUB 1800 ' Get task selection
1250 IF A%=0 THEN RTN%=1:RETURN ' Nothing selected -- return
1260 OPTNO%=A%
1270 IF DATATYPE%<>5 THEN RETURN ' Return if not Ring Seq Chart
1280 R%=TROW%+1:C%=TCOL%:L%=RINGLG%:F$=SPACE$(L%)
1290 LOCATE R%,29:PRINT "Enter Ring Number =>"
1300 LOCATE R%,C%:COLOR FG2%,BG2%:PRINT F$:COLOR FG1%,BG1%
1310 HEXFLD%=1:GOSUB 60000 ' Get ring number
1320 IF F$=SPACE$(L%) THEN RTN%=1:RETURN ' Nothing selected -- return
1330 RINGNO$=F$
1340 RETURN
1398 '
1500 R%=TROW%+2:C%=TCOL%:L%=1:F$=DRATE$ ' Get ring data rate for RSC
1510 LOCATE R%,29:PRINT "Enter Ring Data Rate=> (1= 4, 2= 16)"
1520 LOCATE R%,C%:COLOR FG2%,BG2%:PRINT F$:COLOR FG1%,BG1%
1530 HEXFLD%=1:GOSUB 60000 ' Get user input
1540 IF F$=" " THEN CLS:END ' Nothing entered - end pgm
1550 IF F$<>"1" AND F$<>"2" THEN GOSUB 1710:GOTO 1530 ' Bad selection
1560 IF NEWFILE%=0 AND F$<>DRATE$ THEN DRATE$=F$:GOSUB 7000 ' Save new rate
1570 RETURN
1598 '
1599 '
1600 LOCATE RR%,1:PRINT SPACE$(79);:RETURN ' Clear row RR% on the screen
1699 '
1700 BEEP:LOCATE 23,33:PRINT "CANNOT BE BLANK";:RETURN
1710 BEEP:LOCATE 23,31:PRINT "MUST BE EITHER 1 OR 2";:RETURN
1750 BEEP:LOCATE 21,21:PRINT "NO DATA AVAILABLE FOR SELECTED CHART TYPE":RETURN
1799 '
1800 L%=1:F$=" " ' Initialize variables
1810 LOCATE R%,C%:COLOR FG2%,BG2%:PRINT F$;:COLOR FG1%,BG1% 'Highlight input fld
1820 GOSUB 60000 ' Get user input
1830 IF F$=" " THEN F$="0" ' Nothing entered -- set to zero
1840 IF F$<"0" OR F$>"9" THEN BEEP:GOTO 1820 ' Not numeric -- beep & ask again
1850 A%=VAL(F$) ' Store the numeric value in A%
1860 IF A%<LVAL% OR A%>HVAL% THEN BEEP:GOTO 1820 ' Value out of range -- retry
1870 RETURN
1997 '
1998 '***** SET VARIABLES USED IN THE PROGRAM FOR THE SELECTED CHART TYPE *****
1999 '
2000 ON DATATYPE% GOTO 2010,2020,2030,2035,2040,2050
2010 RESTORE 2100:GOTO 2900
2020 RESTORE 2200:GOTO 2900
2030 RESTORE 2300:GOTO 2900
2035 RESTORE 2350:GOTO 2900
2040 RESTORE 2400:GOTO 2900
2050 RESTORE 2500:GOTO 2900
2099 '
2100 DATA "8228",24,187,1,0,24,0,23,5,0
2110 DATA 3,3,5,5,7,10,10,11,11,12,12,13,13,14,14,15,15,16,16,17,17,20,21
2120 DATA 28,57,32,57,57,27,55,27,55,27,55,27,55,27,55,27,55,27,55,27,55,47,47
2130 DATA 4,8,1,10,4,12,4,12,4,12,4,12,4,12,4,12,4,12,4,14,14
2199 '
2200 DATA "8218",9,63,1,2,9,0,8,3,0
2210 DATA 3,3,4,5,6,7,13,13
2220 DATA 31,50,37,41,41,48,13,57

```

```

2230 DATA 4,4,4,10,8,1,14,14
2299 '
2300 DATA "8219",18,143,1,2,18,0,17,3,0
2310 DATA 3,3,4,5,5,6,6,7,7,9,12,14,17,20,12,14,17
2320 DATA 29,49,42,23,49,25,49,26,55,46,5,5,5,5,63,63,63
2330 DATA 4,4,4,10,10,8,8,1,1,4,14,14,14,1,14,14,14
2349 '
2350 DATA "8220",21,168,1,2,21,0,20,3,0
2360 DATA 3,3,4,4,6,6,7,7,8,8,9,9,10,13,15,18,21,13,15,18
2370 DATA 31,50,27,63,25,50,27,50,23,50,27,57,46,5,5,5,5,63,63,63
2380 DATA 4,4,4,1,10,10,8,8,12,12,1,1,4,14,14,14,1,14,14,14
2399 '
2400 DATA "RING",4,36,1,2,4,0,3,0,0
2410 DATA 11,13,16
2420 DATA 34,34,40
2430 DATA 10,14,6
2499 '
2500 DATA "ADAPTLOC",7,40,1,2,6,7,5,4,1
2510 DATA 12,12,12,12,12
2520 DATA 11,26,42,55,65
2530 DATA 12,8,4,4,4
2599 '
2900 READ DATATYPE$,NOFLDS%,RECLN%,CMP1%,CMP2%,NXT1%,NXT2%,INPFLDS%,HEX1%,HEX2%
2910 FOR I%=1 TO INPFLDS%:READ ROW%(I%):NEXT I%
2920 FOR I%=1 TO INPFLDS%:READ COL%(I%):NEXT I%
2930 TOTLNG%=0
2940 FOR I%=1 TO INPFLDS% '      Save length & location of input flds in rcd
2950 READ FLNG%(I%):FLOC%(I%)=TOTLNG%+1:TOTLNG%=TOTLNG%+FLNG%(I%)
2960 NEXT I%
2965 IF NXT1%>0 THEN FLNG%(NXT1%)=4:FLOC%(NXT1%)=TOTLNG%+1:TOTLNG%=TOTLNG%+4
2966 IF NXT2%>0 THEN FLNG%(NXT2%)=4:FLOC%(NXT2%)=TOTLNG%+1
2967 IF HEX1%>0 THEN FLNG%(HEX1%)=RINGLG% '      Reset length of Ring # field
2970 IF DATATYPE%<>5 THEN EOD%=25 ELSE EOD%=24
2975 BLNK1%=RECLN%-EOD%:BLNK2%=RECLN%-4
2980 RETURN
2997 '
2998 '***** RETRIEVE RECORDS FROM THE DATA FILE *****
2999 '
3100 GET FILE1%,1 '      Read record #1
3110 RECDAT$=BUFFER$ '      Get the record data
3120 LREC=CVS(MID$(RECDAT$,1,4)) '      Get record length
3130 FREC1=CVS(MID$(RECDAT$,5,4)) '      Get 1st record in sort 1
3140 FREC2=CVS(MID$(RECDAT$,9,4)) '      Get 1st record in sort 2
3150 FOPEN=CVS(MID$(RECDAT$,13,4)) '      Get 1st open (deleted) record
3160 ECHN=CVS(MID$(RECDAT$,17,4)) '      Get last rcd in chain
3170 VERSION=CVS(MID$(RECDAT$,21,4)) '      Get file version number
3180 IF DATATYPE%=5 THEN DRATE$=MID$(RECDAT$,EOD%,1)
3190 RETURN
3199 '
3200 RECDAT$=BUFFER$ '      Get the record data
3210 FOR I%=1 TO NOFLDS%:F$(I%)=MID$(RECDAT$,FLOC%(I%),FLNG%(I%)):NEXT I%
3220 RETURN
3299 '
3300 RECDAT$=BUFFER$ '      Get the record data
3310 NOPEN=CVS(MID$(RECDAT$,1,4)) '      Get rcd # of next open rcd
3320 RETURN
3997 '
3998 '***** CHECK WHETHER NEW FILE - IF OLD, OPEN AND READ 1ST RECORD *****
3999 '
4000 NEWFILE%=0 '      Initialize new file flag

```

```

4010 FILENAME$=DATATYPE$ ' Store file name in FILENAME$
4020 IF DATATYPE%=5 THEN FILENAME$=FILENAME$+RINGNO$ ' Add ring no. for RSC
4030 FILENAME$=FILENAME$+".DAT" ' Add file extension to name
4040 OPEN FILENAME$ FOR INPUT AS FILE3% ' Open file to see if it exists
4050 CLOSE FILE3% ' Close the file
4060 IF NEWFILE%=1 THEN GOSUB 4800:RETURN ' If new file, set rcd #1 vars
4070 GOSUB 4900 ' Reopen file as a random file
4080 GOSUB 3100 ' Get rcd #1 from file
4090 IF DATATYPE%=1 AND VERSION<>1 AND VERSION<>2 THEN GOTO 4700
4100 IF DATATYPE%=3 AND VERSION<>2 THEN GOTO 4700
4110 VERSION=2 ' Set current version number
4120 IF ECHN=0 THEN NEWFILE%=1 ' If end of chain=0, file is empty
4130 RETURN
4499 '
4500 ERRNO=ERR ' Routine executed on I/O error
4510 IF ERRNO=53 THEN NEWFILE%=1:RESUME NEXT ' File doesn't exist, set flag
4520 IF ERRNO=61 THEN RESUME 8500 ' Disk full error
4530 IF ERRNO=25 OR ERRNO=27 OR ERRNO=68 THEN RESUME 20500 ' Printer not ready
4540 CLS
4550 LOCATE 12,31:PRINT "WARNING - ERROR ";USING "###";ERRNO ' Other error
4560 PRINT TAB(29) "PROGRAM EXECUTION ENDED"
4570 END
4699 '
4700 CLS:BEEP
4710 LOCATE 12,22:PRINT "DATA FILES NOT CONVERTED TO NEW FORMAT"
4720 LOCATE 13,15:PRINT "Refer to 'Intro and Planning Guide' for instructions"
4730 PRINT:PRINT TAB(29) "PROGRAM EXECUTION ENDED"
4740 END
4799 '
4800 LREC=2:FREC1=2:FREC2=2:FOPEN=0:ECHN=0:VERSION=2:DRATE$=" " ' Init vars
4810 RETURN
4899 '
4900 IF OPEN1%=1 THEN CLOSE FILE1% ' Close the data file if open
4910 OPEN FILENAME$ AS FILE1% LEN=RECLEN% ' Open the file
4920 OPEN1%=1 ' Set the "file open" flag
4930 FIELD FILE1%,RECLEN% AS BUFFER$ ' Define record layout
4940 RECDAT$=SPACE$(RECLEN%) ' Init record buffer variable
4950 RETURN
4997 '
4998 '***** SUBROUTINES USED TO SEARCH FOR RECORD IN FILE *****
4999 '
5000 NEXTR=0:LASTR=0:CMFPLD$="":L%=FLNG%(CMFPLD%) ' Find rcd on 1st sort fld
5010 IF (NXTFLD%=0) OR (NEWFILE%=1) THEN RETURN
5020 WHILE (RECNUM>0) AND (NEXTR=0) AND (SRCHFLD$<>CMFPLD$)
5030 GET FILE1%,RECNUM:GOSUB 3200 ' Get record from data file
5040 CMFPLD$=LEFT$(F$(CMFPLD%),L%) ' Get record comparison field
5050 IF (DATATYPE%>1 AND DATATYPE%<5) AND SRCHFLD$<>CMFPLD$ THEN A$=LEFT$(F$(CM
P2%),L%):IF SRCHFLD$=A$ THEN CMFPLD$=A$
5060 IF SRCHFLD$>CMFPLD$ THEN LASTR=RECNUM:RECNUM=CVS(F$(NXTFLD%))
5070 IF (DATATYPE%=1 OR DATATYPE%=6) AND SRCHFLD$<CMFPLD$ THEN NEXTR=RECNUM
5080 IF (DATATYPE%>1 AND DATATYPE%<6) AND SRCHFLD$<CMFPLD$ THEN LASTR=RECNUM:RE
CNUM=CVS(F$(NXTFLD%))
5090 WEND
5100 IF SRCHFLD$=CMFPLD$ THEN NEXTR=CVS(F$(NXTFLD%))
5110 RETURN
5499 '
5500 CMFPLD$=F$(CMFPLD%) ' Get record comparison field
5510 L%=LEN(CMFPLD$)
5520 WHILE MID$(CMFPLD$,L%,1)=" ":L%=L%-1:WEND ' Find last non-blank character
5530 CMFPLD$=LEFT$(CMFPLD$,L%) ' Remove any trailing blanks

```

```

5540 RETURN
5799 '
5800 NEXTREC=FREC2:RECNUM=0 ' Find rcd preceding CHGREC
5810 WHILE NEXTREC<>CHGREC ' on 2nd sort field
5820 RECNUM=NEXTREC
5830 GET FILE1%,RECNUM:GOSUB 3200
5840 NEXTREC=CVS(F$(NXT2%))
5850 WEND
5860 RETURN
5997 '
5998 '***** DISPLAY INPUT SCREEN FOR RING SEQUENCE CHART OPTIONS *****
5999 '
6000 IF NEWFILE%=1 THEN RINGOPT%=1:RETURN
6010 CLS
6020 LOCATE 3,27:PRINT "RING SEQUENCE CHART OPTIONS"
6030 LOCATE 8,1
6040 PRINT TAB(20) "1) Add component at end of ring"
6050 PRINT TAB(20) "2) Add to ring following selected component"
6060 PRINT TAB(20) "3) Change/delete/display selected component"
6070 R%=CSRLIN+2:C%=53
6080 LOCATE R%,27,1:PRINT "Select an Option (1-3) =>"
6090 GOSUB 6900
6100 LVAL%=0:HVAL%=3:GOSUB 1800:RINGOPT%=A%
6110 IF RINGOPT%=0 THEN ENDCHG%=1
6120 RETURN
6498 '
6499 '
6500 CLS ' Find the specified component
6510 DP$="DP"+SPACE$(FLNG%(1)-2)
6520 F%=1:GOSUB 6700
6599 '
6600 GOSUB 60000:VAR$(F%)=F$ ' Get component # or location
6610 IF F$=B$ THEN ENDCHG%=1:GOTO 6660 ' Nothing entered - return
6620 SRCHFLD$=F$:GOSUB 9000 ' Search for record in file
6630 IF FOUND%=0 THEN BEEP:LOCATE 22,31:PRINT "COMPONENT NOT FOUND":GOTO 6600
6640 IF F%=1 AND F$=DP$ THEN F%=2:SWAP CMP1%,CMP2%:GOSUB 6700:GOTO 6600
6650 GOSUB 9200 ' Store the field values
6660 IF F%=2 THEN SWAP CMP1%,CMP2%
6670 RETURN
6699 '
6700 IF F%=1 THEN R%=12:C%=47:LOCATE R%,21:PRINT "Enter Component Number =>"
6710 IF F%=2 THEN R%=14:C%=49:LOCATE R%,21:PRINT "Enter Component Location =>"
6720 GOSUB 6900
6730 L%=FLNG%(F%):B$=SPACE$(L%):F$=B$
6740 LOCATE R%,C%:COLOR FG2%,BG2%:PRINT B$:COLOR FG1%,BG1%
6750 RETURN
6899 '
6900 LOCATE 24,29:PRINT "or press ENTER to return";:RETURN
6997 '
6998 '***** SAVE THE FIRST RECORD IN THE DATA FILE *****
6999 '
7000 MID$(RECDAT$,1,4)=MKS$(LREC) ' Put record length in record
7010 MID$(RECDAT$,5,4)=MKS$(FREC1) ' Put 1st rcd in sort 1 in record
7020 MID$(RECDAT$,9,4)=MKS$(FREC2) ' Put 1st rcd in sort 2 in record
7030 MID$(RECDAT$,13,4)=MKS$(FOPEN) ' Put 1st open rcd in record
7040 MID$(RECDAT$,17,4)=MKS$(ECHN) ' Put end of chain in record
7050 MID$(RECDAT$,21,4)=MKS$(VERSION) ' Put version number in record
7055 IF DATATYPE%=5 THEN MID$(RECDAT$,EOD%,1)=DRATE$ ' Add ring data rate
7060 MID$(RECDAT$,EOD%+1,BLNK1%)=SPACE$(BLNK1%) 'Blank out rest of record
7070 LSET BUFFER$=RECDAT$ ' Transfer record to buffer area

```

```

7080 PUT FILE1%,1 ' Store record # 1 in file
7090 RETURN
7097 '
7098 '***** Store data record number RECNUM *****
7099 '
7100 FOR I%=1 TO INPFLD$:F$(I%)=VAR$(I%):NEXT I% ' Save input field values
7110 IF NXT1%>0 THEN F$(NXT1%)=MKS$(NEXT1) ' Save ptrs to next rclds in chain
7120 IF NXT2%>0 THEN F$(NXT2%)=MKS$(NEXT2)
7130 FOR I%=1 TO NOFLD$:MID$(RECDAT$,FLOC%(I%),FLNG%(I%))=F$(I%):NEXT I%
7140 LSET BUFFER$=RECDAT$ ' Transfer data to record buffer
7150 PUT FILE1%,RECNUM ' Write buffer into data file
7160 RETURN
7997 '
7998 '***** STORE NEW/CHANGED RCD & RESET LAST RCD # IF NEEDED. *****
7999 '
8000 IF NXT2%>0 THEN GOSUB 9500 ' Find place in file on 2nd sort field
8010 IF NEWFILE%=1 THEN GOSUB 4900:RECNUM=2:NEWFILE%=0:GOTO 8100 ' New file
8020 IF FOUND%=1 THEN RECNUM=CHGREC:GOTO 8100 ' Store rcd in original place
8030 IF FOPEN=0 THEN RECNUM=LREC+1:LREC=RECNUM:GOTO 8100 ' Add a new rcd
8040 RECNUM=FOPEN ' If there's an open rcd, use it
8050 GET FILE1%,FOPEN:GOSUB 3300 ' Get open record from file
8060 FOPEN=NOPEN ' Reset next open rcd # in rcd 1
8099 '
8100 IF NEXT1=FREC1 THEN FREC1=RECNUM ' Reset ptr to 1st rcd in sort order 1
8110 IF NEXT2=FREC2 THEN FREC2=RECNUM ' Reset ptr to 1st rcd in sort order 2
8120 IF NEXT1=0 THEN ECHN=RECNUM ' Reset end of chain rcd # if needed
8130 GOSUB 7000 ' Save 1st record in file
8140 GOSUB 7100 ' Save new/changed record
8150 IF LAST1=0 THEN GOTO 8190 ' Sorted in sort order 1?
8160 GET FILE1%,LAST1:GOSUB 3200 ' Reset ptr of previous rcd in chain
8170 MID$(RECDAT$,FLOC%(NXT1%),FLNG%(NXT1%))=MKS$(RECNUM)
8180 LSET BUFFER$=RECDAT$:PUT FILE1%,LAST1
8190 IF LAST2=0 THEN RETURN ' Sorted in sort order 2?
8200 GET FILE1%,LAST2:GOSUB 3200 ' Reset ptr of previous rcd in chain
8210 MID$(RECDAT$,FLOC%(NXT2%),FLNG%(NXT2%))=MKS$(RECNUM)
8220 LSET BUFFER$=RECDAT$:PUT FILE1%,LAST2
8230 RETURN
8499 '
8500 CLS:BEEP
8510 LOCATE 12,20:PRINT "WARNING - DISK(ETTE) FULL, DATA NOT SAVED"
8520 PRINT TAB(29) "PROGRAM EXECUTION ENDED"
8530 ON ERROR GOTO 0
8540 END
8997 '
8998 '***** SEARCH FOR RECORD IN THE DATA FILE *****
8999 '
9000 FOUND%=0:IF NEWFILE%=1 THEN NEXT1=0:LAST1=0:RETURN
9010 CMPFLD%=CMP1% ' Using sort field 1 for search
9020 RECNUM=FREC1:NXTFLD%=NXT1% ' Save # of 1st rcd in sort order 1
9030 GOSUB 5000 ' Look for the record
9040 NEXT1=NEXTR:LAST1=LASTR ' Save s of previous & next rclds
9050 IF SRCHFLD$=CMPFLD$ THEN FOUND%=1:CHGREC=RECNUM ' If found, set flag
9060 RETURN
9198 '
9199 '
9200 FOR I%=1 TO INPFLD$:VAR$(I%)=LEFT$(F$(I%),FLNG%(I%)):NEXT I%:RETURN
9499 '
9500 CMPFLD%=CMP2$:SRCHFLD$=VAR$(CMP2%) ' Using sort field 2 for search
9510 IF FOUND%=0 THEN GOTO 9600 ' Skip the next lines if new rcd
9520 IF SRCHFLD$=CMPFLD$ THEN RETURN ' Return if 2nd sort fld unchanged

```

```

9530 SAVENXT$=F$(NXT2%) ' Save # of rcd following this one
9540 GOSUB 5800 ' Find record preceding this one
9550 IF CHGREC=FREC2 THEN FREC2=CVS(SAVENXT$):GOTO 9600 ' Reset ptr to 1st rcd
9560 F$(NXT2%)=SAVENXT$ ' Reset rcd ptr in preceding rcd
9570 PUT FILE1%,RECNUM ' Save the preceding rcd
9599 '
9600 RECNUM=FREC2:NXTFLD%=NXT2%
9610 GOSUB 5000 ' Find place in file on 2nd sort fld
9620 IF SRCHFLD$<>CMPFLD$ THEN LAST2=LASTR ELSE LAST2=RECNUM 'Save # of last rcd
9630 NEXT2=NEXTR ' Save # of rcd following it
9640 RETURN
9899 '
9900 FOR I%=1 TO INPFLD$:VAR$(I%)=SPACE$(FLNG%(I%)):NEXT I%:RETURN
9997 '
9998 '***** ADD/CHANGE RECORD IN THE DATA FILE *****
9999 '
10000 RINGOPT%=0:SORTOPT%=1:ENDCHG%=0
10010 ON DATATYPE% GOSUB 51000,52000,53000,53500,54000,55000 'Set up input scrn
10020 IF DATATYPE%=5 THEN GOSUB 6000 ' Get option for Ring Seq Chart
10030 WHILE ENDCHG%=0 ' Loop until end flag is set
10040 ON DATATYPE% GOSUB 11000,11000,11000,11000,12000,13000 'Add/chg/dlet rcd
10050 WEND
10060 RETURN
10398 '
10399 '
10400 LOCATE 24,24:PRINT "Press ENTER to return to main menu";:RETURN
10499 '
10500 R%=24:C%=77:LOCATE R%,4
10510 PRINT "Select 1)CHANGE this entry 2)DELETE this entry 3)DISPLAY new ent
ry => ";
10520 LVAL%=1:HVAL%=3:GOSUB 1800:OPT%=A% ' Get user selection
10530 RR%=24:GOSUB 1600 ' Blank out message area
10540 RETURN
10599 '
10600 R%=24:C%=76:LOCATE R%,6
10610 PRINT "Select 1)SAVE this entry 2)CHANGE this entry 3)SELECT new entry
=> ";
10620 LVAL%=1:HVAL%=3:GOSUB 1800:OPT%=A% ' Get user selection
10630 RR%=24:GOSUB 1600 ' Blank out message area
10640 RETURN
10699 '
10700 F$=VAR$(FLDNO%):L%=FLNG%(FLDNO%):R%=ROW%(FLDNO%):C%=COL%(FLDNO%)
10710 IF FLDNO%=HEX1% OR FLDNO%=HEX2% THEN HEXFLD%=1
10720 GOSUB 60000:VAR$(FLDNO%)=F$
10730 RETURN
10997 '
10998 '***** Change IBM 82xx Chart data *****
10999 '
11000 GOSUB 50000:GOSUB 10400 ' Display the input screen
11010 GOSUB 9900:FLDNO%=1:FOUND%=2 ' Initialize variables
11020 WHILE FOUND%=2 ' Repeat until good unit # entered
11030 GOSUB 10700 ' Get input for first unit #
11040 IF VAR$(1)=SPACE$(FLNG%(1)) THEN ENDCHG%=1:RETURN ' Nothing entered
11050 SRCHFLD$=VAR$(1):GOSUB 11700 ' Search for unit # in 82xx files
11060 IF FOUND%=2 THEN GOSUB 11900 ' Unit # found in other file
11070 WEND
11080 IF FOUND%=1 THEN GOSUB 9000:GOSUB 9200:GOSUB 50500 'Display record
11090 RR%=24:GOSUB 1600 ' Blank out message area
11100 OPT%=0:IF FOUND%=1 THEN GOSUB 10500 ' Ask if user wants to chg/del/dsp
11110 IF OPT%=2 THEN GOSUB 15000:RETURN ' Execute delete subrtne

```



```

11120 IF OPT%=3 THEN RETURN ' Display new entry
11199 '
11200 OPT%=2:IF DATATYPE%=1 THEN NXT%=2 ELSE NXT%=3
11210 WHILE OPT%=2 ' Get inputs for other fields
11220 IF DATATYPE%<>1 THEN GOSUB 11500 ' Get other unit number
11230 FOR FLDNO%=NXT% TO INPFLDS%:GOSUB 10700:NEXT FLDNO% ' Get remaining flds
11240 GOSUB 10600 ' Ask if user wants to save/chg/rtn
11250 GOSUB 9000:IF OPT%=1 THEN GOSUB 8000 ' Store record if save requested
11260 WEND
11270 RETURN
11498 '
11499 '
11500 IF SRCHFLD$=VAR$(1) THEN FLDNO%=2 ELSE FLDNO%=1 ' Check other unit #
11510 SV1$=F$(FLDNO%):SV2$=SRCHFLD$:B$=SPACE$(FLNG$(FLDNO%)):FOUND%=999
11520 WHILE FOUND%>0 ' Check until input is okay
11530 FOUND%=0:GOSUB 10700:F$=VAR$(FLDNO%) ' Get unit number
11540 IF F$=B$ THEN FOUND%=11:GOSUB 1700 ' Display "cannot be blank" msg
11550 IF F$=SV2$ THEN FOUND%=12:GOSUB 11950 ' Display "duplicate unit" msg
11560 IF FOUND%=0 AND F$<>SV1$ THEN SRCHFLD$=F$:GOSUB 11700
11570 IF FOUND%=1 OR FOUND%=2 THEN GOSUB 11900 ' Display "already in file" msg
11580 WEND
11590 RR%=24:GOSUB 1600 ' Blank out message area
11600 SRCHFLD$=SV2$ ' Restore original search value
11610 RETURN
11698 '
11699 '
11700 SAVETYPE%=DATATYPE%:DATATYPE%=1:FOUND%=0:FOUND$=""
11710 WHILE DATATYPE%<5 AND FOUND%=0 ' Look for unit # in all files
11720 CLOSE FILE1%:GOSUB 2000:GOSUB 4000:GOSUB 9000
11730 IF FOUND%=0 THEN DATATYPE%=DATATYPE%+1 ELSE IF DATATYPE%<>SAVETYPE% THEN
FOUND%=2
11740 WEND
11750 IF FOUND%>0 THEN FOUND$=DATATYPE$
11760 DATATYPE%=SAVETYPE%:CLOSE FILE1%:GOSUB 2000:GOSUB 4000
11770 RETURN
11899 '
11900 BEEP:LOCATE 23,21:PRINT "UNIT NUMBER ALREADY DEFINED IN ";FOUND$;" FILE";
11910 RETURN
11950 BEEP:LOCATE 23,26:PRINT "DUPLICATE UNIT NUMBER ENTERED":RETURN
11997 '
11998 '***** Change IBM Ring Sequence Chart data *****
11999 '
12000 IF RINGOPT%<>1 THEN GOSUB 6500:IF ENDCHG%=1 THEN RETURN ' Find component
12010 GOSUB 50000:IF RINGOPT%<>3 THEN GOSUB 10400 ' Display the input screen
12020 ON RINGOPT% GOSUB 12600,12700,12800 ' Insert data on screen
12030 GOSUB 50500 ' Display the input fields
12040 IF RINGOPT%<>3 THEN GOTO 12100
12050 GOSUB 10500 ' Ask if user wants to chg/del/dsp
12060 IF OPT%=2 THEN GOSUB 15000:RETURN
12070 IF OPT%=3 THEN RETURN
12099 '
12100 OPT%=2
12110 WHILE OPT%=2 ' Get inputs for other fields
12120 FOR FLDNO%=1 TO INPFLDS%
12130 GOSUB 10700
12140 IF FLDNO%=1 THEN RR%=24:GOSUB 1600 ' Blank out msg line
12150 IF FLDNO%=1 AND VAR$(1)=SPACE$(FLNG$(1)) THEN ENDCHG%=1:RETURN
12160 NEXT FLDNO%
12170 GOSUB 10600 ' Ask if user wants to save/chg/rtn
12180 IF OPT%=1 THEN GOSUB 8000 ' Save rcd & set flag

```

```

12190 WEND
12200 RETURN
12598 '
12599 '
12600 IF NEWFILE%=0 THEN GET FILE1%,ECHN:GOSUB 3200:LAST1=ECHN:FOUND%=1 ELSE FOU
ND%=0
12610 J%=-7:LST%=INPFLDS%:GOSUB 12900 ' Put preceding record on screen
12620 FOUND%=0:GOSUB 9900:NEXT1=0 ' Initialize input field values
12630 RETURN
12699 '
12700 J%=-7:LST%=INPFLDS%:GOSUB 12900 ' Put preceding record on screen
12710 IF NEXT1>0 THEN GET FILE1%,NEXT1:GOSUB 3200:FOUND%=1 ELSE FOUND%=0
12720 J%=7:LST%=INPFLDS%-1:GOSUB 12900 ' Put following rcd on screen
12730 LAST1=CHGREC ' Save rcd # of preceding rcd
12740 FOUND%=0:GOSUB 9900 ' Initialize input field values
12750 RETURN
12799 '
12800 IF LAST1>0 THEN GET FILE1%,LAST1:GOSUB 3200:FOUND%=1 ELSE FOUND%=0
12810 J%=-7:LST%=INPFLDS%:GOSUB 12900 ' Put preceding record on screen
12820 IF NEXT1>0 THEN GET FILE1%,NEXT1:GOSUB 3200:FOUND%=1 ELSE FOUND%=0
12830 J%=7:LST%=INPFLDS%-1:GOSUB 12900 ' Put following rcd on screen
12840 GET FILE1%,CHGREC:GOSUB 3200 ' Read current record again
12850 FOUND%=1:GOSUB 9200:GOSUB 50500 ' Store old input values & display
12860 RETURN
12899 '
12900 IF FOUND%=0 THEN GOSUB 9900 ELSE GOSUB 9200 ' Store rcd field values
12910 FOR I%=1 TO LST%
12920 LOCATE ROW%(I%)+J%,COL%(I%):PRINT VAR$(I%) ' Put field values on screen
12930 NEXT I%
12940 RETURN
12997 '
12998 '***** Change Locator Chart data *****
12999 '
13000 GOSUB 50000:GOSUB 10400 ' Display the input screen
13010 GOSUB 9900 ' Initialize input fld vars
13020 FLDNO%=1:GOSUB 10700 ' Get input for first field
13030 IF VAR$(1)=SPACE$(FLNG%(1)) THEN ENDCHG%=1:RETURN ' Nothing entered
13040 RR%=24:GOSUB 1600 ' Blank out message area
13050 SRCHFLD$=VAR$(1):GOSUB 9000 ' Search for record in file
13060 IF FOUND%=1 THEN GOSUB 9200:GOSUB 50500 ' Display record if found
13070 OPT%=0:IF FOUND%=1 THEN GOSUB 10500 ' Ask if user wants to chg/del/dsp
13080 IF OPT%=2 THEN GOSUB 15000:RETURN ' Execute delete subrtne
13090 IF OPT%=3 THEN RETURN ' Display new entry
13099 '
13100 OPT%=2
13110 WHILE OPT%=2 ' Get inputs for other fields
13120 FOR FLDNO%=2 TO INPFLDS%:GOSUB 10700:NEXT FLDNO%
13130 GOSUB 10600 ' Ask if user wants to save/chg/rtn
13140 IF OPT%=1 THEN GOSUB 8000 ' Save new/changed rcd & set flag
13150 WEND
13160 RETURN
14997 '
14998 '***** DELETE RECORD FROM THE DATA FILE *****
14999 '
15000 IF LAST1=0 AND NEXT1=0 THEN GOSUB 4800:NEWFILE%=1:CLOSE FILE1%:KILL FILENA
ME$:RETURN
15010 IF NXT2%=0 THEN GOTO 15100 ' Not sorted on 2nd sort field, skip
15020 NEXT2=CVS(F$(NXT2%)) ' Save rcd # of next rcd
15030 GOSUB 5800 ' Find preceding rcd on 2nd sort fld
15040 LAST2=RECNUM ' Save rcd # of preceding record

```

```

15050 IF LAST2=0 AND NEXT2>0 THEN FREC2=NEXT2 ' Reset 1st rcd in sort 2
15099 '
15100 MID$(RECDAT$,1,4)=MK$(FOPEN) ' Put ptr to next open rcd in buffer
15110 MID$(RECDAT$,5,BLNK2%)=SPACE$(BLNK2%) 'Blank out rest of buffer
15120 LSET BUFFER$=RECDAT$
15130 PUT FILE1%,CHGREC ' Store the empty record in the file
15140 FOPEN=CHGREC ' Reset number of the first open rcd
15150 IF LAST1=0 AND NEXT1>0 THEN FREC1=NEXT1 ' Deleted first rcd
15160 IF CHGREC=ECHN AND LAST1>0 THEN ECHN=LAST1 ' Deleted last rcd in chain
15170 GOSUB 7000 ' Store rcd #1 in the file
15199 '
15200 IF LAST1=0 THEN GOTO 15250
15210 GET FILE1%,LAST1:GOSUB 3200 ' Get record preceding deleted one
15220 MID$(RECDAT$,FLOC%(NXT1%),FLNG%(NXT1%))=MK$(NEXT1) 'Save rcd ptr
15230 LSET BUFFER$=RECDAT$ ' Change # of next rcd in chain
15240 PUT FILE1%,LAST1 ' Save record
15250 IF LAST2=0 THEN GOTO 15300
15260 GET FILE1%,LAST2:GOSUB 3200 ' Get record preceding deleted one
15270 MID$(RECDAT$,FLOC%(NXT2%),FLNG%(NXT2%))=MK$(NEXT2) 'Save rcd ptr
15280 LSET BUFFER$=RECDAT$ ' Change # of next rcd in chain
15290 PUT FILE1%,LAST2 ' Save record
15299 '
15300 RETURN
19997 '
19998 '***** PRINT SELECTED CHART *****
19999 '
20000 NXT%=NXT1%:TTIME$=TIME$ ' Initialize sort field & time
20010 ON DATATYPE% GOSUB 21000,21000,21000,21000,22000,23000 ' Print the report
20020 RETURN
20399 '
20400 LOCATE 24,22:PRINT "or press ENTER to return to main menu";:RETURN
20499 '
20500 BEEP
20510 LOCATE 22,16:PRINT "PRINTER ERROR - Prepare printer, then press ENTER";
20520 INPUT "",A$
20530 RETURN
20599 '
20600 LPRINT TAB(42) "Printed on ";DDATE$;" at ";TTIME$:RETURN
20997 '
20998 '***** Print IBM 82xx Cabling Chart *****
20999 '
21000 CLS
21010 LOCATE 3,25:PRINT "IBM ";DATATYPE$;" CABLING CHART OPTIONS"
21020 LOCATE 9,1
21030 PRINT TAB(25) "1) Print all charts for a ring"
21040 PRINT TAB(25) "2) Print chart for one IBM ";DATATYPE$
21050 PRINT TAB(25) "3) Print all IBM ";DATATYPE$;"s"
21060 R%=CSRLIN+2:C%=53
21070 LOCATE R%,27:PRINT "Select an Option (1-3) =>"
21080 GOSUB 20400 ' Display "or press ENTER" msg
21090 LVAL%=0:HVAL%=3:GOSUB 1800:SORTOPT%=A% ' Get option number
21100 IF SORTOPT%=0 THEN RETURN ' Nothing selected -- return
21110 IF SORTOPT%=3 THEN GOSUB 21800:RETURN ' Print entire file
21120 R%=R%+1:LOCATE R%,27
21130 IF SORTOPT%=1 THEN L%=RINGLG%:PRINT "Enter Ring Number =>";
21140 IF SORTOPT%=2 THEN L%=4:PRINT "Enter IBM ";DATATYPE$;" Unit No.=>";
21150 IF SORTOPT%=2 THEN CMPFLD%=1 ELSE IF DATATYPE%=1 THEN CMPFLD%=5 ELSE CMPFLD%=3
21160 FOUND%=0:C%=53:F$=SPACE$(L%)
21199 '

```

```

21200 WHILE FOUND%=0 ' Print chart for ring/unit
21210 LOCATE R%,C%:COLOR FG2%,BG2%:PRINT F$:COLOR FG1%,BG1%
21220 GOSUB 60000:IF F$=SPACE$(L%) THEN RETURN ELSE SRCHFLD$=F$
21230 IF SORTOPT%=1 THEN GOSUB 21300:IF FOUND%=1 THEN RETURN
21240 IF SORTOPT%=2 THEN GOSUB 9000:IF FOUND%=1 THEN GOSUB 21500:RETURN
21250 BEEP:LOCATE R%+2,33
21260 IF SORTOPT%=1 THEN PRINT "NO DATA FOR RING" ELSE PRINT "UNIT NOT FOUND"
21270 WEND
21298 '
21299 '
21300 NEXTREC=FREC1 ' Print all charts for ring
21310 WHILE NEXTREC>0 ' Search entire file
21320 GET FILE1%,NEXTREC:GOSUB 3200
21330 IF LEFT$(F$(CMPFLD%),L%)=SRCHFLD$ THEN FOUND%=1:GOSUB 21500
21340 NEXTREC=CVS(F$(NXT%)) ' Save next rcd #
21350 WEND
21360 RETURN
21499 '
21500 ON DATATYPE% GOSUB 51000,52000,53000,53500 'Set up for selected chart
21510 GOSUB 50700 ' Insert data values in chart
21520 GOSUB 20600:FOR I%=1 TO 5:LPRINT:NEXT I% ' Print date & time line
21530 LPRINT RIGHT$(ROW$(1),LEN(ROW$(1))-4):LPRINT:LPRINT:LPRINT ' Print header
21540 FOR I%=2 TO 22:LPRINT RIGHT$(ROW$(I%),LEN(ROW$(I%))-4):NEXT I%
21550 LPRINT FF$ ' Form feed to skip to new page
21560 RETURN
21799 '
21800 NEXTREC=FREC1 ' Print all charts in file
21810 WHILE NEXTREC>0
21820 GET FILE1%,NEXTREC:GOSUB 3200 ' Read record NEXTREC
21830 GOSUB 21500 ' Print chart for this rcd
21840 NEXTREC=CVS(F$(NXT%)) ' Save pointer to next rcd
21850 WEND
21860 RETURN
21997 '
21998 '***** Print Ring Sequence Chart *****
21999 '
22000 GOSUB 54000 ' Set up print fields
22005 IF DRATE$="1" THEN RT$=" 4" ELSE IF DRATE$="2" THEN RT$="16"
22010 GET FILE1%,ECHN:GOSUB 3200 ' Read last rcd in chain
22020 LINECNT%=0:PAGE%=1:NEXTREC=FREC1 ' Initialize counters
22030 WHILE NEXTREC>0 ' Loop thru all rcds in file
22040 IF LINECNT%=0 THEN GOSUB 22500 ' Print heading lines at top
22050 GET FILE1%,NEXTREC:GOSUB 3200
22060 GOSUB 50800 ' Put field values in rpt lines
22070 FOR I%=10 TO 14:LPRINT ROW$(I%):NEXT I%' Print rpt lines for this comp
22080 NEXTREC=CVS(F$(NXT%)) ' Get rcd # of next component
22090 LINECNT%=LINECNT%+1 ' Increase # of components printed
22100 IF LINECNT%=7 OR NEXTREC=0 THEN GOSUB 22600 ELSE LPRINT ROW$(15):LPRINT R
OW$(16)
22110 WEND
22120 RETURN
22499 '
22500 GOSUB 20600:LPRINT:LPRINT ' Print date & time
22510 LPRINT ROW$(1):LPRINT ' Print page header
22520 LPRINT TAB(10) "Ring Number ";RINGNO$ SPC(10) "Ring Data Rate ";RT$;
22530 LPRINT SPC(12) "Page " USING "##";PAGE%:LPRINT
22540 LPRINT TAB(10) "cable from: ";F$(INPFLD%):LPRINT
22550 RETURN
22599 '
22600 LPRINT:LPRINT SPC(60) "cable to: ";F$(INPFLD%) ' Print page trailer

```

```

22610 LPRINT FF$
22620 LINECNT%=0:PAGE%=PAGE%+1
22630 RETURN
22997 '
22998 '***** Print Locator Chart *****
22999 '
23000 OPT$="Print chart":GOSUB 23700:IF SORTOPT%=0 THEN RETURN ' Get sort optn
23010 GOSUB 55000 ' Set up chart lines
23020 LINECNT%=0
23030 WHILE NEXTREC>0 ' Loop for each rcd in data file
23040 IF LINECNT%=0 THEN GOSUB 23500 ' Print header at top of page
23050 GET FILE1%,NEXTREC:GOSUB 3200
23060 GOSUB 50800 ' Insert field values in lines
23070 LPRINT HDR4$:LPRINT ROW$(ROW%(1))
23080 NEXTREC=CVS(F$(NXT%))
23090 LINECNT%=LINECNT%+1
23100 IF LINECNT%=18 OR NEXTREC=0 THEN GOSUB 23600 ' Print page trailer
23110 WEND
23120 RETURN
23499 '
23500 GOSUB 20600:LPRINT:LPRINT:LPRINT ' Print date & time
23510 IF SORTOPT%=1 THEN LPRINT TAB(23) "ADAPTER ADDRESS TO PHYSICAL LOCATION"
23520 IF SORTOPT%=2 THEN LPRINT TAB(23) "PHYSICAL LOCATION TO ADAPTER ADDRESS"
23530 LPRINT TAB(35) "LOCATOR CHART":LPRINT:LPRINT
23540 LPRINT HDR1$:LPRINT HDR2$:LPRINT HDR3$
23550 RETURN
23599 '
23600 LPRINT HDR6$:LPRINT FF$ ' Print last line of chart
23610 LINECNT%=0 ' Reset counter to start new page
23620 RETURN
23699 '
23700 CLS
23710 LOCATE 3,30:PRINT "LOCATOR CHART OPTIONS"
23720 LOCATE 9,1
23730 PRINT TAB(20) "1) ";OPT$;" in Adapter Address order"
23740 PRINT TAB(20) "2) ";OPT$;" in Physical Location order"
23750 R%=CSRLIN+2:C%=53
23760 LOCATE R%,27:PRINT "Select an Option (1-2) =>"
23770 GOSUB 20400 ' Display "or press ENTER" msg
23780 LVAL%=0:HVAL%=2:GOSUB 1800:SORTOPT%=A% ' Get option number
23790 IF SORTOPT%=1 THEN NXT%=NXT1%:NEXTREC=FREC1 ' Use sort field 1
23800 IF SORTOPT%=2 THEN NXT%=NXT2%:NEXTREC=FREC2 ' Use sort field 2
23810 IF SORTOPT%=2 THEN COL%(2)=COL%(2)-4:SWAP COL%(1),COL%(2)
23820 RETURN
29997 '
29998 '***** DISPLAY ALL FILE ENTRIES ON THE SCREEN *****
29999 '
30000 IF DATATYPE%=1 THEN HDR$=SPACE$(26)+"8228 UNIT NO."+B8$+"RING NO."
30010 IF DATATYPE%=2 THEN HDR$=SPACE$(15)+"8218 UNIT NO. 1"+B8$+"UNIT NO. 2"+B8$
+"RING NO."
30020 IF DATATYPE%=3 THEN HDR$=SPACE$(15)+"8219 UNIT NO. 1"+B8$+"UNIT NO. 2"+B8$
+"RING NO."
30030 IF DATATYPE%=4 THEN HDR$=SPACE$(15)+"8220 UNIT NO. 1"+B8$+"UNIT NO. 2"+B8$
+"RING NO."
30040 IF DATATYPE%=5 THEN HDR$=B14$+"COMPONENT NUMBER"+SPACE$(11)+"LOCATION"+SPA
CE$(11)+"CABLE"
30050 IF DATATYPE%=6 THEN HDR$=B12$+"ADAPTER ADDR PHYS LOC DEV ID RIN
G # UNIT #"
30060 IF DATATYPE%<=6 THEN NXT%=NXT1%:NEXTREC=FREC1 ' Initialize search field
30070 IF DATATYPE%=6 THEN OPT$="List entries":GOSUB 23700:IF SORTOPT%=0 THEN RET

```

```

URN
30080 C%=0 ' Initialize line counter
30099 '
30100 WHILE NEXTREC>0 ' Display all entries in file
30110 GET FILE1%,NEXTREC:GOSUB 3200
30120 IF C%=0 THEN CLS:PRINT HDR$:PRINT STRING$(79,HBAR$)
30130 IF DATATYPE%=1 THEN PRINT TAB(30) F$(1) SPC(15) F$(5)
30140 IF DATATYPE%=2 THEN PRINT TAB(22) F$(1) SPC(14) F$(2) SPC(14) F$(3)
30150 IF DATATYPE%=3 THEN PRINT TAB(22) F$(1) SPC(14) F$(2) SPC(14) F$(3)
30155 IF DATATYPE%=4 THEN PRINT TAB(22) F$(1) SPC(14) F$(2) SPC(14) F$(3)
30160 IF DATATYPE%=5 THEN PRINT TAB(17) F$(1) SPC(11) F$(2) SPC(11) F$(3)
30170 IF DATATYPE%=6 THEN PRINT TAB(12) F$(1) SPC(6) F$(2) SPC(6) F$(3) SPC(6)
F$(4) SPC(6) F$(5)
30180 C%=C%+1:NEXTREC=CVS(F$(NXT%))
30190 IF C%=20 OR NEXTREC=0 THEN LOCATE 24,29,0:PRINT "Press ENTER to continue"
;:INPUT "",A$:C%=0
30200 WEND
30210 RETURN
49997 '
49998 '***** BUILD LINES FOR INPUT SCREEN AND REPORTS *****
49999 '
50000 CLS ' Display input screen
50010 FOR I%=1 TO 24
50020 LOCATE I%,1:PRINT ROW$(I%);
50030 NEXT I%
50040 COLOR FG2%,BG2% ' Put underscores in input flds
50050 FOR I%=1 TO INPFLDS%
50060 LOCATE ROW%(I%),COL%(I%):PRINT SPACES$(FLNG%(I%))
50070 NEXT I%
50080 LOCATE ROW%(1),COL%(1)
50090 COLOR FG1%,BG1%
50100 RETURN
50499 '
50500 COLOR FG2%,BG2% ' Underscore the input fields
50510 FOR I%=1 TO INPFLDS% ' Insert old values in input flds
50520 LOCATE ROW%(I%),COL%(I%):PRINT VAR$(I%)
50530 NEXT I%
50540 COLOR FG1%,BG1%
50550 RETURN
50699 '
50700 GOSUB 9200 ' Put values in rpt flds
50710 FOR I%=1 TO 24:CCNT%(I%)=0:NEXT I%
50720 FOR I%=1 TO INPFLDS% ' Insert ctl chrs to emphasize prt
50730 R%=ROW%(I%)
50731 IF DATATYPE%=2 AND I%=7 THEN R%=R%+1
50732 IF DATATYPE%=3 AND I%=13 THEN R%=R%+1
50733 IF DATATYPE%=3 AND (I%=15 OR I%=16) THEN R%=R%-1
50734 IF DATATYPE%=4 AND I%=16 THEN R%=R%+1
50735 IF DATATYPE%=4 AND (I%=18 OR I%=19) THEN R%=R%-1
50740 R$=ROW$(R%):C%=COL%(I%)+CCNT%(R%)-1:X%=LEN(R$)-C%-FLNG%(I%)
50750 ROW$(R%)=LEFT$(R$,C%)+BH$+VAR$(I%)+EH$
50760 IF X%>0 THEN ROW$(R%)=ROW$(R%)+RIGHT$(R$,X%)
50770 CCNT%(R%)=CCNT%(R%)+4 ' Keep track of extra chrs in line
50780 NEXT I%
50790 RETURN
50799 '
50800 GOSUB 9200 ' Put values in rpt flds
50810 FOR I%=1 TO INPFLDS%
50820 MID$(ROW$(ROW%(I%)),COL%(I%),FLNG%(I%))=VAR$(I%)
50830 NEXT I%

```

```

50840 RETURN
50997 '
50998 '***** 8228 Cabling Chart *****'
50999 '
51000 ROW$(1)=SPACE$(30)+"IBM 8228 CABLING CHART"
51010 ROW$(2)=B4$+"Section 1"
51020 ROW$(3)=B14$+"Unit Number"+B18$+"Location"+B2$
51030 ROW$(4)=B4$
51040 ROW$(5)=B14$+"1= Rack-mounted"+B14$+"Building"+B2$
51050 ROW$(6)=B14$+"2= Wall-mounted"
51060 ROW$(7)=SPACE$(43)+"Ring Number"+B2$
51070 ROW$(8)=B4$+"Section 2"
51080 ROW$(9)=B14$+"Receptacle Connect To"+B16$+"Device"
51090 ROW$(10)=B18$+"1"+SPACE$(39)
51100 ROW$(11)=B18$+"2"+SPACE$(39)
51110 ROW$(12)=B18$+"3"+SPACE$(39)
51120 ROW$(13)=B18$+"4"+SPACE$(39)
51130 ROW$(14)=B18$+"5"+SPACE$(39)
51140 ROW$(15)=B18$+"6"+SPACE$(39)
51150 ROW$(16)=B18$+"7"+SPACE$(39)
51160 ROW$(17)=B18$+"8"+SPACE$(39)
51170 ROW$(18)=B4$
51180 ROW$(19)=B4$+"Section 3"
51190 ROW$(20)=B14$+"A. Connect RI of this 8228 to:"+B16$
51200 ROW$(21)=B14$+"B. Connect RO of this 8228 to:"+B16$
51210 ROW$(22)=B4$
51220 RETURN
51997 '
51998 '***** 8218 Cabling Chart *****'
51999 '
52000 VR1$=VBAR$+B10$+VBAR$:VR2$=UL$+H3$+UR$:VR3$=VBAR$+B3$+VBAR$
52010 VR4$=LL$+HBAR$+TT3$+HBAR$+LR$:VR5$=" Connect to: "
52020 VR6$=STRING$(16,HBAR2$)
52099 '
52100 ROW$(1)=SPACE$(30)+"IBM 8218 CABLING CHART"
52110 ROW$(2)=B4$+"Section 1"
52120 ROW$(3)=SPACE$(36)+"Unit Number"+B2$
52130 ROW$(4)=SPACE$(30)+"Ring"+B2$
52140 ROW$(5)=SPACE$(30)+"Building"+B2$
52150 ROW$(6)=SPACE$(30)+"Location"+B2$
52160 ROW$(7)=SPACE$(30)+"1= Rack-mounted"+B2$
52170 ROW$(8)=SPACE$(30)+"2= Wall-mounted"
52180 ROW$(9)=B4$+"Section 2"
52190 ROW$(10)=SPACE$(28)+UL$+H6$+UR$+B10$+UL$+H6$+UR$
52200 ROW$(11)=SPACE$(28)+VBAR$+B4$+"RI"+VR1$+"RI"+B4$+VBAR$
52210 ROW$(12)=SPACE$(13)+VR5$+VBAR$+VR2$+" "+VR1$+VR2$+" "+VBAR$+VR5$
52220 ROW$(13)=SPACE$(28)+VBAR$+VR3$+" "+VR1$+VR3$+" "+VBAR$+B2$
52230 ROW$(14)=SPACE$(28)+VBAR$+VR4$+" "+VR1$+VR4$+" "+VBAR$
52240 ROW$(15)=B12$+VR6$+CR3$+HBAR2$+HBAR2$+LR2$+" RO"+VR1$+"RO"+LL2$+STRING$(3,
HBAR2$)+CR3$+VR6$
52250 ROW$(16)=SPACE$(28)+VBAR$+VR2$+" "+VR1$+VR2$+" "+VBAR$+B4$+" Yellow"
52260 ROW$(17)=SPACE$(28)+VBAR$+VR3$+" "+VR1$+VR3$+" "+VBAR$+" Crossover Cable"
52270 ROW$(18)=SPACE$(28)+VBAR$+VR4$+" "+VR1$+VR4$+" "+VBAR$
52280 ROW$(19)=SPACE$(28)+VBAR$+B2$+LL2$+STRING$(3,HBAR2$)+CR3$+STRING$(10,HBAR2
$)+CR3$+HBAR2$+HBAR2$+LR2$+B3$+VBAR$
52290 ROW$(20)=SPACE$(28)+LL$+H6$+LR$+" Yellow "+LL$+H6$+LR$
52300 ROW$(21)=SPACE$(34)+"Crossover Cable"
52310 ROW$(22)=B4$
52320 RETURN
52997 '

```

```

52998 '***** 8219 Cabling Chart *****
52999 '
53000 VR1$=" "+B6$:VR2$=" _ Rx ":VR3$=" Tx ":VR4$="( ) ":VR5$=" \ "
53010 VR6$=VBAR$+VR5$+VBAR$:VR7$=VBAR$+" \ B"+VBAR$:VR8$=VBAR$+" \ 0"+VBAR$
53020 VR9$=" Connect to: ":VR10$=STRING$(23,HBAR2$)
53030 VR11$=UL$+HBAR$+HBAR$+UR$:VR12$=VBAR$+B2$+VBAR$:VR13$=LL$+HBAR$+TT3$+LR$
53099 '
53100 ROW$(1)=SPACE$(30)+"IBM 8219 CABLING CHART"
53110 ROW$(2)=B4$+"Section 1"
53120 ROW$(3)=SPACE$(35)+"Unit Number"+B2$
53130 ROW$(4)=SPACE$(35)+"Ring"+B2$
53140 ROW$(5)=SPACE$(36)+"Building"+B2$
53150 ROW$(6)=SPACE$(36)+"Location"+B2$
53160 ROW$(7)=SPACE$(27)+"1=Rack mount 2=Wall mount"+B2$
53170 ROW$(8)=B4$+"Section 2"
53180 ROW$(9)=SPACE$(31)+"Cable Number"+B2$
53190 ROW$(10)=SPACE$(27)+UL$+H8$+UR$+VR5$+UL$+H8$+UR$
53200 ROW$(11)=SPACE$(27)+VBAR$+VR1$+VR6$+VR1$+VBAR$+B12$
53210 ROW$(12)=B18$+"0-0"+UR$+" 0"+UL$+CR$+VR4$+B4$+VR7$
53215 ROW$(12)=ROW$(12)+VR4$+H4$+CR$+UR$+B4$+UL$+"0-0"
53220 ROW$(13)=SPACE$(21)+RT$+H4$+LT$+VBAR$+VR2$+VR6$+VR2$+VBAR$+RT$+H4$+LT$+B6$
53230 ROW$(14)=B18$+"B-B"+LR$+" B"+LL$+CR$+VR4$+B4$+VR8$
53235 ROW$(14)=ROW$(14)+VR4$+H4$+CR$+LR$+B4$+LL$+"B-B"
53240 ROW$(15)=SPACE$(27)+VBAR$+VR3$+VR6$+VR3$+VBAR$
53250 ROW$(16)=B6$+VR9$+B8$+VBAR$+VR11$+B4$+VR6$+VR11$+B4$+VBAR$+B8$+VR9$
53260 ROW$(17)=SPACE$(27)+VBAR$+VR12$+B4$+VR6$+VR12$+B4$+VBAR$+B10$
53270 ROW$(18)=SPACE$(27)+VBAR$+VR13$+B4$+VR6$+VR13$+B4$+VBAR$
53280 ROW$(19)=B4$+VR10$+CR3$+HBAR2$+HBAR2$+LR2$+" "+B4$+VR6$+B2$+LL2$+STRING$(5
,HBAR2$)+CR3$+VR10$
53290 ROW$(20)=B6$+"1= Yellow Crossover "+LL$+H8$+LR$+VR5$+LL$+H8$+LR$+B4$+"Ye
low Crossover"
53300 ROW$(21)=B14$+"Cable"+SPACE$(43)+"Cable"
53310 ROW$(22)=B6$+"2= Patch Cable"
53320 RETURN
53497 '
53498 '***** 8220 Cabling Chart *****
53499 '
53500 VR1$=" "+B6$:VR2$=" _ Rx ":VR3$=" Tx ":VR4$="( ) ":VR5$=" \ "
53510 VR6$=VBAR$+VR5$+VBAR$:VR7$=VBAR$+" \ B"+VBAR$:VR8$=VBAR$+" \ 0"+VBAR$
53520 VR9$=" Connect to: ":VR10$=STRING$(23,HBAR2$)
53530 VR11$=UL$+HBAR$+HBAR$+UR$:VR12$=VBAR$+B2$+VBAR$:VR13$=LL$+HBAR$+TT3$+LR$
53599 '
53600 ROW$(1)=SPACE$(30)+"IBM 8220 CABLING CHART"
53610 ROW$(2)=B4$
53620 ROW$(3)=B4$+"Section 1"+SPACE$(23)+"Unit Number"+B2$
53630 ROW$(4)=SPACE$(20)+"Ring# "+B10$+"Ring Data Rate (1=4 2=16)" +B2$
53640 ROW$(5)=B14$+"RI (Upstream) RI/RO Switch Settings RO (Downstream)"
53650 ROW$(6)=SPACE$(37)+"Building"+B4$
53660 ROW$(7)=SPACE$(37)+"Location"+B4$
53670 ROW$(8)=SPACE$(37)+"Address"+B6$
53680 ROW$(9)=SPACE$(29)+"1=Rack mount 2=Wall mount"+B2$
53690 ROW$(10)=B4$+"Section 2"+B18$+"Cable Number"+B2$
53700 ROW$(11)=SPACE$(27)+UL$+H8$+UR$+VR5$+UL$+H8$+UR$
53710 ROW$(12)=SPACE$(27)+VBAR$+VR1$+VR6$+VR1$+VBAR$+B12$
53720 ROW$(13)=B18$+"0-0"+UR$+" 0"+UL$+CR$+VR4$+B4$+VR7$
53725 ROW$(13)=ROW$(13)+VR4$+H4$+CR$+UR$+B4$+UL$+"0-0"
53730 ROW$(14)=SPACE$(21)+RT$+H4$+LT$+VBAR$+VR2$+VR6$+VR2$+VBAR$+RT$+H4$+LT$+B6$
53740 ROW$(15)=B18$+"B-B"+LR$+" B"+LL$+CR$+VR4$+B4$+VR8$
53745 ROW$(15)=ROW$(15)+VR4$+H4$+CR$+LR$+B4$+LL$+"B-B"
53750 ROW$(16)=SPACE$(27)+VBAR$+VR3$+VR6$+VR3$+VBAR$

```



```

53760 ROW$(17)=B6$+VR9$+B8$+VBAR$+VR11$+B4$+VR6$+VR11$+B4$+VBAR$+B8$+VR9$
53770 ROW$(18)=SPACE$(27)+VBAR$+VR12$+B4$+VR6$+VR12$+B4$+VBAR$+B10$
53780 ROW$(19)=SPACE$(27)+VBAR$+VR13$+B4$+VR6$+VR13$+B4$+VBAR$
53790 ROW$(20)=B4$+VR10$+CR3$+HBAR2$+HBAR2$+LR2$+" "+B4$+VR6$+B2$+LL2$+STRING$(5
,HBAR2$)+CR3$+VR10$
53800 ROW$(21)=B6$+"1= Yellow Crossover "+LL$+H8$+LR$+VR5$+LL$+H8$+LR$+B4$+"Ye
low Crossover"
53810 ROW$(22)=B6$+"2= Patch Cable"+SPACE$(42)+"Cable"
53820 ROW$(23)=B4$
53830 RETURN
53997 '
53998 '***** Ring Sequence Chart *****
53999 '
54000 GOSUB 54300
54010 ROW$(1)=SPACE$(30)+"RING SEQUENCE CHART"
54020 ROW$(2)=""
54030 ROW$(3)=BOX1$
54040 ROW$(4)=BOX2$
54050 ROW$(5)=BOX3$
54060 ROW$(6)=BOX4$
54070 ROW$(7)=BOX5$
54080 ROW$(8)=CBL1$
54090 ROW$(9)=CBL2$
54100 ROW$(10)=BOX1$
54110 ROW$(11)=BOX2$
54120 ROW$(12)=BOX3$
54130 ROW$(13)=BOX4$
54140 ROW$(14)=BOX5$
54150 ROW$(15)=CBL1$
54160 ROW$(16)=CBL2$
54170 ROW$(17)=BOX1$
54180 ROW$(18)=BOX2$
54190 ROW$(19)=BOX3$
54200 ROW$(20)=BOX4$
54210 ROW$(21)=BOX5$
54220 ROW$(22)=""
54230 RETURN
54299 '
54300 BOX1$=B18$+UL$+H4$+BT$+HBAR$+HBAR$+BT$+STRING$(33,HBAR$)+UR$
54310 BOX2$=B18$+VBAR$+"(Component)"+SPACE$(30)+VBAR$
54320 BOX3$=B18$+VBAR$+STRING$(41,HBAR$)+VBAR$
54330 BOX4$=B18$+VBAR$+"(Location)"+SPACE$(31)+VBAR$
54340 BOX5$=B18$+LL$+STRING$(33,HBAR$)+TT$+HBAR$+HBAR$+TT$+H4$+LR$
54350 CBL1$=SPACE$(23)+UL$+H6$+SPACE$(19)+H3$+LR$+B2$+VBAR$
54360 CBL2$=SPACE$(23)+VBAR$+B2$+UL$+H3$+" Cable "+B12$+H6$+LR$
54370 RETURN
54997 '
54998 '***** Locator Chart *****
54999 '
55000 ON SORTOPT% GOSUB 55300,55500:GOSUB 55700
55010 ROW$(1)=SPACE$(34)+"LOCATOR CHART"
55020 FOR I%=2 TO 6:ROW$(I%)="":NEXT I%
55070 ROW$(7)=HDR1$
55080 ROW$(8)=HDR2$
55090 ROW$(9)=HDR3$
55100 ROW$(10)=HDR4$
55110 ROW$(11)=HDR5$
55120 ROW$(12)=HDR5$
55130 ROW$(13)=HDR5$
55140 ROW$(14)=HDR6$

```

```

55150 FOR I%=15 TO 22:ROW$(I%)="":NEXT I%
55230 RETURN
55299 '
55300 VR1$=H14$:VR2$=H10$:VR3$=B14$:VR4$=B10$
55310 VR5$=" Adapter"+B6$:VR6$=" Address"+B6$
55320 VR7$=" Physical ":VR8$=" Location "
55330 RETURN
55499 '
55500 VR1$=H10$:VR2$=H14$:VR3$=B10$:VR4$=B14$
55510 VR5$=" Physical ":VR6$=" Location "
55520 VR7$=" Adapter"+B6$:VR8$=" Address"+B6$
55530 RETURN
55699 '
55700 HDR1$=B8$+UL$+VR1$+TT$+VR2$+TT$+H16$+TT$+H8$+TT$+H10$+UR$
55710 HDR2$=B8$+VBAR$+VR5$+VBAR$+VR7$+VBAR$+" Device          "+VBAR$+" Ring  "+
VBAR$+" IBM 8228 "+VBAR$
55720 HDR3$=B8$+VBAR$+VR6$+VBAR$+VR8$+VBAR$+" Identification "+VBAR$+" Number "+
VBAR$+" Number  "+VBAR$
55730 HDR4$=B8$+RT$+VR1$+CR$+VR2$+CR$+H16$+CR$+H8$+CR$+H10$+LT$
55740 HDR5$=B8$+VBAR$+VR3$+VBAR$+VR4$+VBAR$+B16$+VBAR$+B8$+VBAR$+B10$+VBAR$
55750 HDR6$=B8$+LL$+VR1$+BT$+VR2$+BT$+H16$+BT$+H8$+BT$+H10$+LR$
55760 RETURN
59997 '
59998 '***** EDIT USER ENTRIES IN INPUT SCREEN FIELDS *****
59999 '
60000 QUIT%=0:0%=0:HEXERR%=0
60010 WHILE QUIT%=0
60020 LOCATE R%,C%+0%,1:C$="":FLG%=0
60030 WHILE LEN(C$)=0:C$=INKEY$:WEND:A%=ASC(C$)
60040 IF A%=13 THEN QUIT%=1:FLG%=1
60050 IF A%>31 AND A%<127 AND 0%<L% THEN FLG%=2:IF A%>96 AND A%<123 THEN A%=A%-
32
60060 IF 0%>0 AND (A%=8 OR (A%=0 AND ASC(RIGHT$(" "+C$,1))=75)) THEN 0%=0%-1:FL
G%=1
60070 IF 0%<L% AND (A%=0 AND ASC(RIGHT$(" "+C$,1))=77) THEN 0%=0%+1:FLG%=1
60080 IF FLG%=2 THEN COLOR FG2%,BG2%:PRINT C$;:COLOR FG1%,BG1%:0%=0%+1:MID$(F$,
0%,1)=CHR$(A%)
60090 RR%=23:GOSUB 1600
60100 IF FLG%=0 THEN IF 0%=L% THEN BEEP ELSE GOSUB 60900
60110 WEND
60120 IF F$<>SPACE$(L%) THEN WHILE LEFT$(F$,1)=" ":F$=RIGHT$(F$,L%-1)+" ":WEND
60130 IF HEXFLD%=1 THEN GOSUB 60500:IF HEXERR%=1 THEN GOSUB 60910:GOTO 60000
60140 HEXFLD%=0:LOCATE 1,1,0
60150 RETURN
60499 '
60500 FOR I%=1 TO L%
60510 A%=ASC(MID$(F$,I%,1))
60520 IF A%<>32 AND (A%<48 OR (A%>57 AND A%<65) OR A%>70) THEN HEXERR%=1
60530 NEXT I%
60540 RETURN
60899 '
60900 BEEP:LOCATE 23,36:PRINT "INVALID KEY";:RETURN
60910 BEEP:LOCATE 23,28:PRINT "MUST BE HEXADECIMAL VALUE";:RETURN

```

Sample Program Listing for CVTFILES

```

2  '***** Convert LAN Files to New Format *****'
3  '
5  NEWVER=2
10 DATA "8228",177,187,13,10,164
20 DATA "8219",139,143,50,4,89
99 '

100 ON ERROR GOTO 10000 '           Branch to 5000 if error
110 CLS:PRINT "Converting Token-Ring Network files to new format . . ."
120 FILE2$="$STRNET$.TMP" '         Save name of temporary file
130 FILE1$="8228.DAT":RESTORE 10:GOSUB 1000 ' Convert 8228 file, if needed
140 FILE1$="8219.DAT":RESTORE 20:GOSUB 1000 ' Convert 8219 file, if needed
150 ON ERROR GOTO 0 '              Turn off error trapping
160 END
999 '

1000 NEWFILE%=0 '                  Initialize "new file" flag
1010 OPEN FILE1$ FOR INPUT AS #3 '  Open file to see if it exists
1020 CLOSE #3 '                    Close the file
1030 IF NEWFILE%=1 THEN RETURN '   If flag is on, check next file
1499 '

1500 READ TYPE$,LEN1%,LEN2%,FF%,FM%,FL% ' Get type, old/new LRECL, flds
1510 B1%=LEN1%-24:B2%=LEN2%-24 '     # of blanks at end of 1st rcd
1510 OPEN FILE1$ AS #1 LEN=LEN1% '   Open data file as a random file
1520 FIELD #1,4 AS LREC1$,12 AS DATA1$,4 AS ECHN1$,4 AS VERSION1$,B1% AS BLNK1$
1530 GET #1,1 '                    Get record #1
1540 ECHN=CVS(ECHN1$) '            Get last rcd in chain
1550 IF ECHN=0 THEN CLOSE #1:KILL FILE1$:RETURN ' Delete file if empty
1560 VERSION=CVS(VERSION1$) '      Get current version number
1570 IF TYPE$="8228" AND (VERSION=1 OR VERSION=NEWVER) THEN CLOSE #1:RETURN
1580 IF TYPE$="8219" AND VERSION=NEWVER THEN CLOSE #1:RETURN
1999 '

2000 OPEN FILE2$ AS #2 LEN=LEN2% '  Open the temporary file
2010 FIELD #2,4 AS LREC2$,12 AS DATA2$,4 AS ECHN2$,4 AS VERSION2$,B2% AS BLNK2$
2020 LSET LREC2$=LREC1$ '          Store # records in buffer
2030 LSET DATA2$=DATA1$ '        Store other fields
2040 LSET ECHN2$=ECHN1$ '        Store last rcd in chain
2050 LSET VERSION2$=MK$$(NEWVER) ' Put new version no. in buffer
2060 LSET BLNK2$=SPACE$(B2%) '    Blank out rest of buffer
2070 PUT #2,1 '                   Store rcd #1 in temp file
2499 '

2500 LREC=CVS(LREC1$) '           Get # of records in file
2510 FIELD #1,FF% AS F1A$,FL% AS F1C$
2520 FIELD #2,FF% AS F2A$,FM% AS F2B$,FL% AS F2C$
2530 PAD$=SPACE$(FM%)
2540 FOR I%=2 TO LREC '           Copy rcds from old file to new
2550 GET #1,I%
2560 LSET F2A$=F1A$
2570 LSET F2B$=PAD$
2580 LSET F2C$=F1C$
2590 PUT #2,I%
2600 NEXT I%
2610 CLOSE #1,#2
2620 KILL FILE1$ '                Erase old data file
2630 NAME FILE2$ AS FILE1$ '      Rename temp file
2650 PRINT:PRINT TYPE$+" FILE converted to new format"
2660 RETURN
9997 '
9998 '***** Error Routine *****'

```

```
9999 '
10000 ERRNO=ERR ' Routine executed on I/O error
10010 IF ERRNO=53 THEN NEWFILE%=1:RESUME NEXT ' File doesn't exist, set flag
10020 CLS
10030 LOCATE 12,31:PRINT "WARNING - ERROR ";USING "###";ERRNO ' Other error
10040 ON ERROR GOTO 0 ' Turn off error trapping
10050 END
```