TSKST1 -- Keyboard SET Command   MACRO V05.05   Thursday 19-Jan-89 14:57
Table of contents

```
    1                                                    .TITLE   TSKST1 -- Keyboard SET Command routines
    2                                                    .ENABL   LC
    3                                                    .DSABL   GBL
    4 000000                                             .CSECT   TSKST1
    5 000000                               TSKST1:
    6                                      ;
    7                                      ;    TSKST1 is the portion of TSKMON that contains the code
    8                                      ;    to implement the SET command.
    9                                      ;
   10                                      ;    Copyright 1978,1979,1980,1981,1982,1983,1984,1985,1986,1987,1988,1989.
   11                                      ;    S&H Computer Systems, Inc,
   12                                      ;    Nashville, Tennessee
   13                                      ;
   14                                      ;    Macro calls
   15                                      ;
   16                                              .MCALL   .CSISPC,.TTOUTR,.SRESET
   17                                              .MCALL   .READW,.TTYIN,.TTYOUT,.PURGE
   18                                              .MCALL   .CSIGEN,.SAVEST,.REOPEN
   19                                              .MCALL   .GTLIN,.GTIM,.DATE,.SPFUN
   20                                              .MCALL   .PRINT,.CLOSE,.LOOKUP
   21                                              .MCALL   .WRITW,.ENTER,.EXIT
   22                                              .MCALL   .SERR,.HERR,.FPROT,.GVAL,.PVAL,.DSTAT
   23                                      ;
   24                                      ;    Global definitions
   25                                      ;
   26                                              .GLOBL   TSKST1,CMDHD,CMDOFF,KDOCIN,SKPSPC,UCLCMD
   27                                              .GLOBL   DORUN,CMDFRM,CMDDSN,STLGCN,DATTIM,PRGALL
   28                                              .GLOBL   DLCEMT,ALCDEV,CMDSHO,CMDSET,CMDWHO,CMDMEM,CMDUSE
   29                                      ;
   30                                      ;    Global references
   31                                      ;
   32                                              .GLOBL   LSW11,$PWKEY,SETCL,SETHST,DVFLAG,DX$NST,TM$MRS
   33                                              .GLOBL   SC$SUC,SC$WRN,SC$ERR,SC$SEV,SC$FTL,SC$UNC,SC$NON
   34                                              .GLOBL   SETPRC,SETTTY,STVRFY,STNOVR,PO$SYS,RCLREV,PA$FLG,PA$DTS
   35                                              .GLOBL   PA$GRC,PA$UKC,PA$DSC,PA$BLD,PA$ULN,PA$DWD,PA$HQL,PA$LET
   36                                              .GLOBL   EM$PTA,EM$PTU,SBPSUF,CHKCLU,EM$CLX,PA$BEL,EM$OPR,PA$NWD
   37                                              .GLOBL   SCNOPS,TM$XBK,CL$XLN,CKCLUS,ACRSTR,R50KMN,SJEMT,RJEMT
   38                                              .GLOBL   PEKEMT,PEKADR,PEKSIZ,TM$NNR,CDBUF,CDGET,TM$IN1,TM$IN2
   39                                              .GLOBL   SYPSWD,EM$SPL,ACRTXT,JPWDEV,JPWTYP,JPWFLG
   40                                              .GLOBL   EM$WCO,EM$WC1,EM$WC2,EM$WC3,EM$WCM,P2$CGR
   41                                              .GLOBL   CXTRMN,R$CHN,R$XCHN,CHNSIZ,C.USED,CL$EPN,CL$EPS,CLEOFS
   42                                              .GLOBL   EDMALD,EM$STL,EM$IST,CLSFEP,EM$IDR,VPRIDF,SETWRD
   43                                              .GLOBL   CLRPRV,OPTLST,PFSO,PFCO,PVNPW,PO$SPV,PRIVSO,PRVOPT
   44                                              .GLOBL   TM$PVA,TM$PVC,PRIVAO,EM$CNO,EM$CPO,EM$CAP,RSTPRV
   45                                              .GLOBL   CHKEQ,CKACOJ,PO$OPR,CKSYPV,P2$TRM,PRIVC2,PO$NAM,EM$NPR
   46                                              .GLOBL   INSTBL,INGADR,INGEMT,IIBUF,II$NAM,II$FLG,II$$SZ,EM$NAD
   47                                              .GLOBL   INSTBN,AF$SCA,AF$NOW,AF$MEM,PO$DBG,PRIVCO,PRVLST
   48                                              .GLOBL   ABRTAD,ABRTCD,CINFLG,$VNOTT,II$PRV,II$NPV
   49                                              .GLOBL   TM$RD1,TM$RD2,TM$LCL,TM$GBL,SPACE1,RC$OWN,RC$CNT
   50                                              .GLOBL   RC$EXC,RC$AGE,RC$AEP,RC$USE,RC$FLG,RC$GBL,RC$NAM
   51                                              .GLOBL   RC$LEN,RC$PVT,RCBBAS,RCBEND,RC$$SZ,SHRRCB,SHRRCN
   52                                              .GLOBL   LP$SPD,LP$PAR,LP$ODD,LP$7BT,EM$ICL,PROSLT,RC$LCG
   53                                              .GLOBL   EM$NPD,EM$ILN,EM$CIP,EM$NSF,EM$IUN,EM$CLN
   54                                              .GLOBL   EM$ILN,EM$ACL,EM$TSL,EM$CLB,EM$NSL,EM$SLT,EM$SLW
   55                                              .GLOBL   SLKDON,SLKDOF,EM$UIO,TM$PR1,TM$PR2,TM$LPR,TM$HPR
   56                                              .GLOBL   TM$HPE,TM$CNG,TM$CDS,TM$CEN,TRMHD1,TRMHD2,AT$DEV
   57                                              .GLOBL   OPRTXT,CLLINE,LCLTXT,REMTXT,TM$AUT,CLFREE,CLUNIT
```

```
 58                            .GLOBL    TM$CLO, TM$CL1, TM$CL2, TM$CL3, TM$CL4, TM$CL5, TM$CL6
 59                            .GLOBL    QHDMS1, QHDMS2, DVSHH1, DVSHH2, DVSHH3, SYASHD, DKASHD
 60                            .GLOBL    TM$NAD, ALCHD1, ALCHD2, TM$NSD, TM$SDN, LNAME
 61                            .GLOBL    CORUSR, LSW, $CTRLO, SERFLG, IOABFL, $CHACT, $STSNG
 62                            .GLOBL    LSTHL, LCLUNT, FSTIOL, LSTIOL, CL$LIX, CW$PRO, CONFG2
 63                            .GLOBL    CL$RQH, CL$WQH, MAXALC, ALCTBL, ALCEND
 64                            .GLOBL    AD$DVU, AD$JOB, AD$$SZ, UCIDEF, HANCHN
 65                            .GLOBL    NEDCHR, LOUTIR, LINIR, LINRTS, CLOTIR
 66                            .GLOBL    CO$DEF, CL$COL, LCDTYP, SOPALC, SOPDAT, SOPTIM
 67                            .GLOBL    UTRPAD, JSWLOC, ERRLOC, MAXMEM, MAXPRI
 68                            .GLOBL    USRSTK, $KINIT, CFSTK, MXJMEM, DFJMEM, EM$HNI
 69                            .GLOBL    SPUBUF, SXBPNT, MXJADR
 70                            .GLOBL    TMTOTH, TMTOTL, TMUSRH, TMIOWH, LDMNT, EM$CSE
 71                            .GLOBL    TMSWTH, TMIDLH, TMIOH, TMSWPH, LDCLEN
 72                            .GLOBL    WILDFL, $NOIN, $NOWTT, $HITTY
 73                            .GLOBL    TECO, EDIT, KED, K52, $1STLG, $DIBOL
 74                            .GLOBL    SH$VAL, SH$NAM, SH$$SZ, SH$RTN, SH$FLG
 75                            .GLOBL    SO$NVL, SO$OCT, SO$NO, HANENT, HANSIZ
 76                            .GLOBL    H.CSR, H.VEC, DVSTAT, SFID, ACRSPD, HANPAR, LSTSPL
 77                            .GLOBL    HAZEL, HAZLFL, HAZLNO, $MLOCK, MDT, GETKCH
 78                            .GLOBL    LINBUF, LINNXT, LSTACT, PRGTOP, PRGSIZ, KMNHI
 79                            .GLOBL    KMNTOP, KMNPGS, KMNSTK, KMNSTR, CXTPAG, FSTIOL
 80                            .GLOBL    LINPNT, LINCNT, LACTIV, LRDTIM, CS$RON
 81                            .GLOBL    LOTBUF, LOTNXT, LOTPNT, $VTESC
 82                            .GLOBL    LOTSIZ, LOTSPC, LCOL, $SLKED, ESC, VDBFLG
 83                            .GLOBL    LAFSIZ, LFWLIM, LINCUR, NUMON, ILSW2
 84                            .GLOBL    $DBKMN
 85                            .GLOBL    $CARUP, DOASGN, UKMNAM, $UKMON, LSW9
 86                            .GLOBL    LSUCF, $CCLRN, VLDSYS, EM$NUK, S$QMIO
 87                            .GLOBL    KL3CLR, $PRGLK, LSW5, PVON, S$SPND, $AUTO
 88                            .GLOBL    S$TWFN, S$TTFN, S$OTFN, S$IOFN, S$OTLO
 89                            .GLOBL    LSTDL, FSTDL, $DETCH, UMSYTP, S$TTSC
 90                            .GLOBL    $DISCN, LPROJ, LPROG, LUNAME, S$RT, S$LOW
 91                            .GLOBL    LCPUHI, LCPULO, LCONTM, $CTRLS, $SPLJB, TXTCL
 92                            .GLOBL    STPFLG, TOTON, USPLCH, SPLCHN, S$HICP
 93                            .GLOBL    S$INWT, S$OTWT, S$TMWT, S$SFWT, S9600
 94                            .GLOBL    S$MSWT, CFBUF, CFEND, CCLSAV, KMNCHN
 95                            .GLOBL    MINTIM, LSECPT, MAXSEC, $EMTTR, VCSHNB
 96                            .GLOBL    OKFILE, OKFEND, $CLTST, UCISPC, MHNSIZ
 97                            .GLOBL    CASTBR, CASCBR, CASTBW, CASCUP, MHNSMS
 98                            .GLOBL    CASTRO, CASTWO, CLTOTL, CO$DTR, CLSFSP
 99                            .GLOBL    CO$CR, CO$FF, CO$FFO, CO$LC, CO$TAB, CO$CTL
100                            .GLOBL    CO$LFI, CO$LFO, CO$BNI, CO$BNO, CL$OPT
101                            .GLOBL    CL$LEN, CL$SKP, CL$WID, CL$LIN, PHYMEM
102                            .GLOBL    LJSW, CTRLTT, NEWJSW, JSTKND, VIMAGE
103                            .GLOBL    USTART, GENTOP, BOTDEV, BOTUNI, CSHALC
104                            .GLOBL    $CTRLC, LSW2, $INKMN, CHAIN, UFORM
105                            .GLOBL    $SGQO, $SGQ3, LITIME
106                            .GLOBL    MAXASN, $CFABT, INDSTA, INDERR
107                            .GLOBL    RUNDEV, LNBLKS, CXTBAS, CXTWDS, UHIMEM
108                            .GLOBL    $DILUP, CSHDEV, CSHDVN, LNSBLK
109                            .GLOBL    LSW3, LSW2S, $DUPRN
110                            .GLOBL    $FORM, $TAB, LSCCA, $CFSOT, LOFSPC, R50COM
111                            .GLOBL    $PAGE, $SCOPE, $ECHO, $LC, $8BIT, CHKALC
112                            .GLOBL    UCHAN, $FORMO, $CFALL, $CFDCC, $CFCCL
113                            .GLOBL    LNPRIM, LNMAP, CW$50H, CONFIG, $SUCF
114                            .GLOBL    $DOOFF, NUCHN, LRBFIL, CFIND, TALEMT
```

```
115          .GLOBL   C.CSW, C.DEVQ, C.SBLK, NLINES, CO$8BT
116          .GLOBL   CD$NAM, CD$DVU, CD$BAS, CD$JOB, CD$$SZ, CD$$UB
117          .GLOBL   LTSCMD, LNSPAC, CFNEST, UCLNAM
118          .GLOBL   $CFOPN, CFSEND, PBFEND, CFSP, $TTGAG
119          .GLOBL   UFPTRP, SDSFCB, SD$DEL, CFLFL4, $UCLCF
120          .GLOBL   SDFLAG, SD$FLK, SD$WFM, SDFORM, $UCLRN
121          .GLOBL   SDBUF1, SDBLK, NSPLDV, LD$RON, $UCLCM, $UCLCL
122          .GLOBL   LDNAME, LDSIZE, LDFLAG, LDBASE, LDPDEV
123          .GLOBL   LSW8, $SGQ1, $SGQ1A, $SGQ1B, $SGQ1C, $SGQ2, $SGIIO, $SGHIO
124          .GLOBL   $DEFER, CFCHAN, SCHAIN, LDDEVX, $SGALL
125          .GLOBL   CFPNT, CFBLK, $QUIET, DIABFL
126          .GLOBL   DIABNO, VT52NO, LA36NO, LA36FL
127          .GLOBL   LSW4, KL4CLR, SDSKIP, SDBU, SD$BAK
128          .GLOBL   $INCOR, $KED, VQUN1B, VINTIO, VQUN1C
129          .GLOBL   SF$BSY, SFFORM, SD$SNG, SFNMBL, NFRESB
130          .GLOBL   SD$HLD, SF$HLD, CURPRM, PRMPNT, SF$1ST
131          .GLOBL   LSTPRM, PRMBUF, PRMEND, CFSPND
132          .GLOBL   SDFHD, SFFLAG, SFQLNK, CFHOLD, LOGDVU, LOGBAS
133          .GLOBL   LCOL, $QTSET, $TECO, CD$TOP, LOGCHK
134          .GLOBL   $WILD, ERRSEV, UERSEV, PASLIN, LOGBAS, LOGDVU
135          .GLOBL   LSTPL, SDCB, SDCBND, VQUANO, VQUAN3
136          .GLOBL   VQUAN1, VQUN1A, VQUAN2, VHIPCT, VQUANO, VQUAN3
137          .GLOBL   DCTRD, DCCRD, DCTWR, DCCWR, ASNSRC
138          .GLOBL   VCORTM, NUMDCD, VNUMDC, KMPRMT, MXPRMT
139          .GLOBL   RDB, RDBEND, RT$NAM, RT$$SZ, CLDEVX, SDDVU
140          .GLOBL   SDNAME, SDCBSZ, LSTSL, LSTATE
141          .GLOBL   TK1VAL, CINDAT, SYSDAT, SYTIMH, SYTIML
142          .GLOBL   BASMAP, LOMAP, HIMAP, JCXPGS
143          .GLOBL   SMRSIZ, SRTSIZ, CSHSIZ, TK1SEC
144          .GLOBL   TSXLN, TSXSIT, GRT1, TRGRET, LICTXT, SUPCOD, NAMTOP, SUMS, SUCS
145          .GLOBL   LPRG1, LPRG2, S$QUSR, S$IOWT, S$SFWT
146          .GLOBL   S$SPDB, S$SPCB, SFUSER, SFFILE, VT200, VT2007, VT2008
147          .GLOBL   LCBIT, LA36, LA120, VT52, VT100, DIABLO, QUME
148          .GLOBL   ADM3A, LTRMTP, LA12FL, LA12NO, VT52FL
149          .GLOBL   VT10FL, VT10NO, QUMEFL, QUMENO, ADM3FL
150          .GLOBL   VT20FL, VT20NO
151          .GLOBL   ADM3NO, SYINDX, SYUNIT, NUMDEV, PNAME
152          .GLOBL   OF$DEV, OF$UNT, OF$FIL, OF$FLG, SYNAME
153          .GLOBL   OF$$SZ, OT$RON, RESDEV, $TAPE
154          .GLOBL   KMNBAS, ODTBAS, $CTRLD
155          .GLOBL   LSW6, $SNWTT, PF$SYS, PF$IOW, $DEBUG
156          .GLOBL   RSR, TSR, LMXNUM, LSTMX, MXDTR, ZCLR, MXCSR
157          .GLOBL   $INDDF, $INDRN, IN$ACT, IN$CNT, IN$CMD, INDSAV
158          .GLOBL   $PHONE, INVEC, LMXLN, MXVEC, $INIT, $DEAD, $HARD
159          .GLOBL   ITRMTP, LMXPRM, LSW7, $INDAB, CFSTS, CF$IND, CF$QUT
160          .GLOBL   CFABLV, MONVEC, LBSPRI, MAXPRI, MXJPRI, LPRI, $SYSPS
161          .GLOBL   LOGCHN, LOGFLG, LOGPTR, LOGBUF, LOGBLK, LF$IN, LF$OUT
162          .GLOBL   LF$OPN, LF$WRT, UCLBLK, UCLDAT
163          .GLOBL   CSHHD, FC$CDX, FC$LNK, FD$NAM, UC$NDC, UC$MDC, CVTUC
164          .GLOBL   CMDBUF, PAUMSG, RDCMD, DKSAV, SYSAV, CVTTAB, RUNHD, SEARCH
165          .GLOBL   INVOPT, FKILL, ABRTCF, ACRFN, XAREA, FILNAM, NOPRG, FPRINT
166          .GLOBL   PUSHCF, TRMSTR, FILNAM, R50DIR, R50SY, R50IND, R50SAV
167          .GLOBL   INDACT, R50DUP, R50PIP, R50KED, R50K52, R50KEX, R50TSX, R50UCL
168          .GLOBL   BLKO, RDERM, R50VIR, NOSTRT, RUNEMT, OVRCOR
169          .GLOBL   BADSAV, LDNAM, NOPRG, NOCIN, SIZVAL, ASKLNM, BADCMD, KCSIBF
170          .GLOBL   ASDEX, KCSIMS, ASNOVF, GTRD50, R50BUF, R50LDO, MNTDEV, DMTARG
171          .GLOBL   DEADEV, CHKMNT, CHKMTX, INFOMT, NOFLAG, MTOPHD, INVOPT, ILLCMD
```

```
172          . GLOBL   R5OLD, INVLDN, R5ODSK, ACRFIL, BDFNAM, LOGASN, MNTFUL, R5OLD7
173          . GLOBL   TBLOVF, SETHD, CSIMS2, CKPRIV, R5ONO, AMBOPT, ACRDEC
174          . GLOBL   MAXAVL, PRTDEC, DEVUNT, PNAME, HNBUF, CKTERM
175          . GLOBL   ACROCT, HANBSY, CSIMS1, MISSEQ, NOIND, POPCF
176          . GLOBL   BADPMT, BADPRI, TOTXT, CRLF, HIPRI, STLGHD, LOGCLS, R5OLOG
177          . GLOBL   BDLGOP, SPLHLA, NOCCL, LDOPHD, PRTFIX, PRTSPC
178          . GLOBL   DLTXT, OCTFIX, PRTTTP, NATXT, SPDTX1, NOTXT, YESTXT, NINTXT
179          . GLOBL   PRTUNM, SYHD1, SYHD2, PRTLN, SPACE2, DETTXT, SPACE3, RNMS
180          . GLOBL   SWPTX, LOCKTX, SPACE5, PRTDC3, KBMSG, DIVIDE, PRTDC2
181          . GLOBL   COLOO, CPUAH, CPUAL, PRTTMV, NOFIL, CMDBUF, CALUCL
182          . GLOBL   NOUDC, DEVHD1, ASNHD1, ASNHD2, SHMTH1, SHMTH2, PRTTMD
183          . GLOBL   CVDVNM, SPACE6, PRTBUF, PRTFNM, NONEMS, NODAT, NOLDMT
184          . GLOBL   SUBARO, EDTFIL, RONTXT, NOTAVL, KBTX, MNFLGS, MNBPC
185          . GLOBL   DELSPC, MNBASE, MNTOP, MONHD, MONAR1, NOPMGN, PMBUSY, MONAR2
186          . GLOBL   NSWPMS, MAXMTX, CURMTX, CHKDLM, SPLHD, AMBOPT, INVOPT
187          . GLOBL   DEVIDL, COAL, ALDEX, COAD, SPACTV, SPWFM, DEVIDL, SPSNG
188          . GLOBL   COAL, ALDEX, ALDBLK, COAD, SPACTV, SPWFM, DEVIDL
189          . GLOBL   SPSNG, SPFUL, SPCF, SPFLK, NOFIL, SPGEMT, NOOPTT
190          . GLOBL   BDLIN, MSGBUF, MSGEND, NOTON, GAGMSG
191          . GLOBL   LINFRE, DJABMS, DLMSG, INVTIM, DMTALL
192          . GLOBL   SHTMSG, AUTHFN, SPLACT, DOSTOP, OFFEMT, KILEMT, UPTMMS
193          . GLOBL   TMTOTH, DIVSOR, TMTOTL, PRTPCT, SUM1, SUM2, SUM3, SUM4
194          . GLOBL   SUM5, SUM6, SUM7, OTHRON, SPLPND, STPASK, SRTSMS, CHKTTD
195          . GLOBL   SIZEMT, ASNOVF, INVLDM, CSIMS4, MNTARG, HUPARG, R5OTT
196          . GLOBL   KMNNAM, NOKMON, CCLNAM, OTRMNT, CHKDEV, DMTSUB, CMDCCL
197          . GLOBL   SHOHD, SUBTXT, MNTTXT, SRTTXT, TOTMMS, UMSSMS, SSRMAP
198          . GLOBL   TSXSMS, USRMMS, JCXSMS, DZTXT, OCTPRT
199          . GLOBL   PRTR50, PRTDAT, PRTTOD, PRTTIM, INVDEV, ALFN, R5ODK
200          . GLOBL   DETHD, DETARG, RUNMS, NOFRDL, R5OMON, INVDAT, MUL32, COAF
201          . GLOBL   AR$PRJ, AR$PRG, AR$CON, AR$CNT, AR$CPH, AR$CPL, AR$UNM
202          . GLOBL   AR$DMY, AR$$SZ, ARNRPB, $SLON, $SLTTY, $SLLET
203          . GLOBL   PRTWRN, SLMXLN, VSLEDT, $LOFCF, CSHMSG
204          . GLOBL   AF$HIE, AF$NOI, $NOINT, AF$PLK, AF$DBG, DS$DIR
205          . GLOBL   AF$IOP, $RNIOP, VOFFTM, VONTM, VTMOUT, VTMIN, VTMLOC, VUSPHN
```

```
     1                                      ;
     2                                      ;   Assembly constants
     3                                      ;
     4          000012                      LF      =       12              ;LINE FEED
     5          000015                      CR      =       15              ;CARRIAGE RETURN
     6          000040                      BLANK   =       40              ;ASCII SPACE
     7          000007                      BELL    =       07              ;ASCII BELL
     8          000011                      TAB     =       11              ;HORIZONTAL TAB
     9          000014                      FF      =       14              ;FORM FEED
    10          000054                      COMMA   =       54              ;COMMA
    11          000400                      BLKWDS  =       256.            ;# OF WORDS IN DISK BLOCK
    12          132500                      WLDNAM  =       132500          ;RAD50 /*/ (WILDCARD)
```

```
 1                                      ;------------------------------------------------------
 2                                      ;   Macro to cause a fatal error message to be printed.
 3                                      ;
 4                                              .MACRO  FERR    MSG
 5                                              MOV     R5,-(SP)
 6                                              MOV     MSG,R5
 7                                              CALL    FPRINT
 8                                              MOV     (SP)+,R5
 9                                              .ENDM   FERR
10                                      ;
11                                      ;------------------------------------------------------
12                                      ;   Macro to print a fatal error message, clean up
13                                      ;   and then jump to RDCMD.
14                                      ;
15                                              .MACRO  FABORT  MSG
16                                              MOV     MSG,R5
17                                              JMP     FKILL
18                                              .ENDM   FABORT
19                                      ;
20                                      ;------------------------------------------------------
21                                      ;   Macro to print a warning message
22                                      ;
23                                              .MACRO  FWARN   MSG
24                                              MOV     R5,-(SP)
25                                              MOV     MSG,R5
26                                              CALL    PRTWRN
27                                              MOV     (SP)+,R5
28                                              .ENDM   FWARN
29                                      ;
30                                      ;------------------------------------------------------
31                                      ;   Macro to start a standard option table.
32                                      ;   Name = 1 to 4 character table name.
33                                      ;   NA = Number of arguments per table entry.
34                                      ;
35                                              .MACRO  TBLDEF  NAME,NA
36                                      NARGS   =       NA
37                                              .CSECT  CMDVS1
38                                      NAME'HD: .WORD  2*NA
39                                              .ENDM   TBLDEF
40                                      ;
41                                      ;------------------------------------------------------
42                                      ;   Macro to enter an option text name and a set of parameters
43                                      ;   into the currently open table.
44                                      ;   STRNG = Ascii name
45                                      ;   A,B,C = Set of option parameters to store in table with name.
46                                      ;
47                                              .MACRO  CMDDEF  STRNG,A,B,C
48                                              .CSECT  NAMES1
49                                      L       =       .
50                                              .ASCIZ  /STRNG/
51                                              .CSECT  CMDVS1
52                                              .WORD   L                       ;POINTER TO NAME STRING
53                                              .WORD   A
54                                      .IIF    GE,<NARGS-2>    .WORD   B
55                                      .IIF    GE,<NARGS-3>    .WORD   C
56                                              .ENDM   CMDDEF
57                                      ;
```

```
    58                                ;--------------------------------------------------------------------
    59                                ;  Macro to end a set of table entries.
    60                                ;
    61                                        .MACRO   TBLEND
    62                                        .CSECT   CMDVS1
    63                                        .WORD    0
    64                                        .CSECT   TSKST1
    65                                        .ENDM    TBLEND
```

```
 1                                          ; ------------------------------------------------------------------
 2                                          ; Define options for the SET command.
 3                                          ;
 4                                          ; Table of "device" names for the SET command and a
 5                                          ; pointer to a set of sub-options for the device.
 6                                          ;
 7                                          ;
 8 000000                                           TBLDEF  SET, 3
 9 000002                                           CMDDEF  CA*CHE, STCHNB, 0, 0
10 000012                                           CMDDEF  CC*L, SETSUB, CCLHD, 0
11 000022                                           CMDDEF  CO*RTIM, SETSVL, VCORTM, 10000.
12 000032                                           CMDDEF  CTRLD, SETSUB, CTDHD, 0
13 000042                                           CMDDEF  ED*IT, SETSUB, EDITHD, 0
14 000052                                           CMDDEF  EM*T, SETSUB, EMTHD, 0
15 000062                                           CMDDEF  ENDS*TARTUP, SETESU, 0, 0
16 000072                                           CMDDEF  ER*ROR, SETSUB, ERRHD, 0
17 000102                                           CMDDEF  EX*IT, RDCMD, 0, 0
18 000112                                           CMDDEF  HIP*RCT, SETSVL, VHIPCT, 10000.
19 000122                                           CMDDEF  HO*ST, SETHST, 0, 0
20 000132                                           CMDDEF  IND, SETSUB, INDHD, 0
21 000142                                           CMDDEF  INT*IOC, SETSVL, VINTIO, 10000.
22 000152                                           CMDDEF  IO*ABORT, SETPRV, IOABHD, 0
23 000162                                           CMDDEF  KMO*N, SETSUB, KMONHD, 0
24 000172                                           CMDDEF  LAN*GUAGE, SETSUB, LANGHD, 0
25 000202                                           CMDDEF  LOG, SETLOG, 0, 0
26 000212                                           CMDDEF  LOGO*FF, SETLOF, 0, 0
27 000222                                           CMDDEF  MAX*PRIORITY, SETMPR, 0, 0
28 000232                                           CMDDEF  NUM*DC, SETNDC, NUMDCD, 0
29 000242                                           CMDDEF  OFF*TIM, SETSVL, VOFFTM, 10000.
30 000252                                           CMDDEF  ONT*IM, SETSVL, VONTM, 10000.
31 000262                                           CMDDEF  PHO*NE, SETSVB, VUSPHN, 1
32 000272                                           CMDDEF  PROC*ESS, SETPRC, 0, 0
33 000302                                           CMDDEF  PROM*PT, SETPRT, 0, 0
34 000312                                           CMDDEF  PRI*ORITY, SETPRI, 0, 0
35 000322                                           CMDDEF  PRINTS*CREEN, SETPRS, 0, 0
36 000332                                           CMDDEF  PRINTW*INDOW, SETPRS, 0, 0
37 000342                                           CMDDEF  QUANO, SETSVL, VQUANO, 10000.
38 000352                                           CMDDEF  QUAN1, SETSVL, VQUAN1, 10000.
39 000362                                           CMDDEF  QUAN1A, SETSVL, VQUN1A, 10000.
40 000372                                           CMDDEF  QUAN1B, SETSVL, VQUN1B, 10000.
41 000402                                           CMDDEF  QUAN1C, SETSVL, VQUN1C, 10000.
42 000412                                           CMDDEF  QUAN2, SETSVL, VQUAN2, 10000.
43 000422                                           CMDDEF  QUAN3, SETSVL, VQUAN3, 10000.
44 000432                                           CMDDEF  REC*ALL, SETREC, 0, 0
45 000442                                           CMDDEF  SHUT*DOWN, SETSHT, 0, 0
46 000452                                           CMDDEF  NOSHUT*DOWN, SETNSH, 0, 0
47 000462                                           CMDDEF  SIG*NAL, SETSUB, SIGHD, 0
48 000472                                           CMDDEF  SL, SETSL, 0, 0
49 000502                                           CMDDEF  SUB*PROCESS, SETSBP, 0, 0
50 000512                                           CMDDEF  SYSP*ASSWORD, SETSYP, 0, 0
51 000522                                           CMDDEF  SYSP*WD, SETSYP, 0, 0
52 000532                                           CMDDEF  TE*RMINAL, SETTTY, 0, 0
53 000542                                           CMDDEF  TT*Y, SETTTY, 0, 0
54 000552                                           CMDDEF  TIMI*N, SETSVL, VTMIN, 10000.
55 000562                                           CMDDEF  TIML*OC, SETSVL, VTMLOC, 10000.
56 000572                                           CMDDEF  TIMO*UT, SETSVL, VTMOUT, 10000.
57 000602                                           CMDDEF  UC*L, SETSUB, UCLHD, 0
```

```
        58 000612                          CMDDEF   US*R,RDCMD,0,0
        59 000622                          CMDDEF   VER*IFY,STVRFY,0,0
        60 000632                          CMDDEF   NOVER*IFY,STNOVR,0,0
        61 000642                          CMDDEF   WI*LDCARDS,SETSUB,WILDHD,0
        62 000652                          CMDDEF   WIN*DOWS,SETWIN,0,0
        63 000662                          CMDDEF   WINP*RT,SETPRS,0,0
        64 000672                          TBLEND
```

```
     1                                    ;
     2                                    ;   Define options for SET EDIT command
     3                                    ;
     4 000000                                    TBLDEF   EDIT,3
     5 000676                                    CMDDEF   ED*IT,SEEDIT,0
     6 000706                                    CMDDEF   KED,SEKED,0
     7 000716                                    CMDDEF   K52,SEK52,0
     8 000726                                    CMDDEF   T*ECO,SETECO,0
     9 000736                                    TBLEND
    10                                    ;
    11                                    ;   Define options for SET ERROR command
    12                                    ;
    13 000000                                    TBLDEF   ERR,2
    14 000742                                    CMDDEF   W*ARNING,SETSEV,SC$WRN
    15 000750                                    CMDDEF   E*RROR,SETSEV,SC$ERR
    16 000756                                    CMDDEF   S*EVERE,SETSEV,SC$SEV
    17 000764                                    CMDDEF   F*ATAL,SETSEV,SC$FTL
    18 000772                                    CMDDEF   U*NCONDITIONAL,SETSEV,SC$UNC
    19 001000                                    CMDDEF   N*ONE,SETSEV,SC$NON
    20 001006                                    TBLEND
    21                                    ;
    22                                    ;   Define options for SET WILDCARDS command
    23                                    ;
    24 000000                                    TBLDEF   WILD,3
    25 001012                                    CMDDEF   IM*PLICIT,SETBIT,LSW5,$WILD
    26 001022                                    CMDDEF   EX*PLICIT,CLRBIT,LSW5,$WILD
    27 001032                                    TBLEND
    28                                    ;
    29                                    ;   Define options for the SET KMON command
    30                                    ;
    31 000000                                    TBLDEF   KMON,3
    32 001036                                    CMDDEF   IND,SETIND,LSW5,$INDDF
    33 001046                                    CMDDEF   NOI*ND,RSTIND,LSW5,$INDDF
    34 001056                                    CMDDEF   SY*STEM,CLRBIT,LSW7,$UKMON
    35 001066                                    CMDDEF   UCI,SETUKM,0,0
    36 001076                                    CMDDEF   USE*R,SETUKM,0,0
    37 001106                                    CMDDEF   NEW,SETKNW,0,0
    38 001116                                    CMDDEF   DEBUG,SETBIT,LSW9,$DBKMN
    39 001126                                    CMDDEF   NODEBUG,CLRBIT,LSW9,$DBKMN
    40 001136                                    TBLEND
    41                                    ;
    42                                    ;   Define options for the SET CCL command
    43                                    ;
    44 000000                                    TBLDEF   CCL,3
    45 001142                                    CMDDEF   T*EST,SETBIT,LSW5,$CLTST
    46 001152                                    CMDDEF   NOT*EST,CLRBIT,LSW5,$CLTST
    47 001162                                    CMDDEF   NEW,SETCNW,0,0
    48 001172                                    TBLEND
    49                                    ;
    50                                    ;   Define options for the SET UCL command
    51                                    ;
    52 000000                                    TBLDEF   UCL,3
    53 001176                                    CMDDEF   F*IRST,STUCLF,LSW7,$UCLCF
    54 001206                                    CMDDEF   M*IDDLE,STUCLM,LSW7,$UCLCM
    55 001216                                    CMDDEF   L*AST,STUCLL,LSW7,$UCLCL
    56 001226                                    CMDDEF   N*ONE,STUCLN,LSW7,0
    57 001236                                    TBLEND
```

```
 58                                          ;
 59                                          ;   Define options for the SET IND command
 60                                          ;
 61 000000                                           TBLDEF   IND,3
 62 001242                                           CMDDEF   AB*ORT,SETBIT,LSW7,$INDAB
 63 001252                                           CMDDEF   NOA*BORT,CLRBIT,LSW7,$INDAB
 64 001262                                           TBLEND
 65                                          ;
 66                                          ;   Define options for the SET LD command
 67                                          ;
 68 000000                                           TBLDEF   LDOP,1
 69 001266                                           CMDDEF   CL*EAN,SLDCLN
 70 001272                                           CMDDEF   EMP*TY,SLDEMP
 71 001276                                           CMDDEF   FR*EE,SLDFRE
 72 001302                                           CMDDEF   WR*ITE,SLDWRT
 73 001306                                           CMDDEF   NOWR*ITE,SLDNWR
 74 001312                                           TBLEND
 75                                          ;
 76                                          ;   Define options for the SET SL command
 77                                          ;
 78 000000                                           TBLDEF   SLE,1
 79 001316                                           CMDDEF   AS*K,SLONOP
 80 001322                                           CMDDEF   KED,SLOKED
 81 001326                                           CMDDEF   K52,SLOKED
 82 001332                                           CMDDEF   KEX,SLOKED
 83 001336                                           CMDDEF   NOKED,SLORT
 84 001342                                           CMDDEF   LEA*RN,SLOUNI
 85 001346                                           CMDDEF   NOLEA*RN,SLONOP
 86 001352                                           CMDDEF   LET,SLOLET
 87 001356                                           CMDDEF   NOLET,SLONLT
 88 001362                                           CMDDEF   OF*F,SLOOFF
 89 001366                                           CMDDEF   ON,SLOON
 90 001372                                           CMDDEF   KM*ON,SLOON
 91 001376                                           CMDDEF   RT*11,SLORT
 92 001402                                           CMDDEF   SUB*STITUTE,SLOLET
 93 001406                                           CMDDEF   NOSUB*STITUTE,SLONLT
 94 001412                                           CMDDEF   SYS*GEN,SLONOP
 95 001416                                           CMDDEF   TT*YIN,SLOTTY
 96 001426                                           CMDDEF   NOTT*YIN,SLONTT
 97 001426                                           CMDDEF   WID*TH,SLOWID
 98 001432                                           CMDDEF   REC*ALL,SLONOP
 99 001436                                           CMDDEF   NOREC*ALL,SLONOP
100 001442                                           TBLEND
101                                          ;
102                                          ;   Define options for the SET RECALL
103                                          ;
104 000000                                           TBLDEF   RCL,1
105 001446                                           CMDDEF   N*ORMAL,0
106 001452                                           CMDDEF   R*EVERSE,1
107 001456                                           TBLEND
108                                          ;
109                                          ;   Define options for the SET WINDOW command
110                                          ;
111 000000                                           TBLDEF   WIN,2
112 001462                                           CMDDEF   COL*UMNS,WINCOL,0
113 001470                                           CMDDEF   DA*RK,WINBIT,WFNORM
114 001476                                           CMDDEF   DEL*ETE,WINBIT,WFDEL
```

```
115 001504                                    CMDDEF    LI*GHT,WINBIT,WFREV
116 001512                                    CMDDEF    NAR*ROW,WINBIT,WF80
117 001520                                    CMDDEF    NOR*MAL,WINBIT,WFNORM
118 001526                                    CMDDEF    OFF,WINBIT,WFDEL
119 001534                                    CMDDEF    ON,WINBIT,0
120 001542                                    CMDDEF    REV*ERSE,WINBIT,WFREV
121 001550                                    CMDDEF    SCR*OLL,WINSCR,0
122 001556                                    CMDDEF    NOSCR*OLL,WINNOS
123 001564                                    CMDDEF    WI*DE,WINBIT,WF132
124 001572                                    TBLEND
125                          ;
126                          ; Define options for the SET PRINTWINDOW command
127                          ;
128 000000                                    TBLDEF    PRS,1
129 001576                                    CMDDEF    BEL*L,PRSBEL
130 001602                                    CMDDEF    NOBEL*L,PRSNBL
131 001606                                    CMDDEF    DEV*ICE,PRSDEV
132 001612                                    CMDDEF    DR*AFT,PRSDRF
133 001616                                    CMDDEF    FL*AG,PRSFLG
134 001622                                    CMDDEF    NOFL*AG,PRSNFL
135 001626                                    CMDDEF    KEY*PRINT,PRSKEY
136 001632                                    CMDDEF    NOKEY*PRINT,PRSNKY
137 001636                                    CMDDEF    LET*TERQUALITY,PRSLET
138 001642                                    CMDDEF    ST*AMP,PRSSTM
139 001646                                    CMDDEF    NOST*AMP,PRSNST
140 001652                                    CMDDEF    TY*PE,PRSTYP
141 001656                                    CMDDEF    WID*TH,PRSWID
142 001662                                    CMDDEF    NOWID*TH,PRSNWD
143 001666                                    TBLEND
144                          ;
145                          ; Printer types for SET PRINTWINDOW/TYPE=type
146                          ;
147 000000                                    TBLDEF    PWT,2
148 001672                                    CMDDEF    AS*CII,0,0
149 001700                                    CMDDEF    FOR*EIGN,0,0
150 001706                                    CMDDEF    PRINT*RONIX,0,0
151 001714                                    CMDDEF    SIM*PLE,0,0
152 001722                                    CMDDEF    ST*ANDARD,0,0
153 001730                                    CMDDEF    LA36,0,0
154 001736                                    CMDDEF    LA120,0,0
155 001744                                    CMDDEF    LA100,1,PA$GRC!PA$UKC!PA$ULN!PA$HQL!PA$DWD
156 001752                                    CMDDEF    LA12,1,PA$GRC!PA$UKC!PA$BLD!PA$ULN!PA$HQL!PA$DWD
157 001760                                    CMDDEF    LA50,1,PA$GRC!PA$UKC!PA$DSC!PA$BLD!PA$ULN!PA$HQL!PA$DWD
158 001766                                    CMDDEF    LA210,1,PA$GRC!PA$UKC!PA$DSC!PA$BLD!PA$ULN!PA$HQL!PA$DWD
159 001774                                    CMDDEF    LQPO2,1,PA$BLD!PA$ULN
160 002002                                    CMDDEF    LQPO3,1,PA$BLD!PA$ULN
161 002010                                    CMDDEF    LNO3,1,PA$GRC!PA$BLD!PA$ULN!PA$DWD
162 002016                                    TBLEND
163                          ;
164                          ; Define options for the SET LANGUAGE command
165                          ;
166 000000                                    TBLDEF    LANG,3
167 002022                                    CMDDEF    DI*BOL,SETBIT,LSW6,$DIBOL
168 002032                                    CMDDEF    DBL,CLRBIT,LSW6,$DIBOL
169 002042                                    TBLEND
170                          ;
171                          ; Define options for the SET EMT command
```

```
172                                                  ;
173 000000                                                   TBLDEF    EMT, 3
174 002046                                                   CMDDEF    TR*ACE, SETEMT, LSW6, $EMTTR
175 002056                                                   CMDDEF    NOTR*ACE, CLRBIT, LSW6, $EMTTR
176 002066                                                   TBLEND
177                                                  ;
178                                                  ;  Define options for the SET IOABORT command
179                                                  ;
180 000000                                                   TBLDEF    IOAB, 3
181 002072                                                   CMDDEF    AB*ORT, STRWRD, IOABFL, 1
182 002102                                                   CMDDEF    NOAB*ORT, STRWRD, IOABFL, 0
183 002112                                                   TBLEND
184                                                  ;
185                                                  ;  Define options for the SET SIGNAL command
186                                                  ;
187 000000                                                   TBLDEF    SIG, 3
188 002116                                                   CMDDEF    H*IPRCT, SETBIT, LSW8, $SGHIO
189 002126                                                   CMDDEF    NOH*IPRCT, CLRBIT, LSW8, $SGHIO
190 002136                                                   CMDDEF    I*NTIOC, SETBIT, LSW8, $SGIIO
191 002146                                                   CMDDEF    NOI*NTIOC, CLRBIT, LSW8, $SGIIO
192 002156                                                   CMDDEF    QUANO, SETBIT, LSW8, $SGQO
193 002166                                                   CMDDEF    NOQUANO, CLRBIT, LSW8, $SGQO
194 002176                                                   CMDDEF    QUAN1, SETBIT, LSW8, $SGQ1
195 002206                                                   CMDDEF    NOQUAN1, CLRBIT, LSW8, $SGQ1
196 002216                                                   CMDDEF    QUAN1A, SETBIT, LSW8, $SGQ1A
197 002226                                                   CMDDEF    NOQUAN1A, CLRBIT, LSW8, $SGQ1A
198 002236                                                   CMDDEF    QUAN1B, SETBIT, LSW8, $SGQ1B
199 002246                                                   CMDDEF    NOQUAN1B, CLRBIT, LSW8, $SGQ1B
200 002256                                                   CMDDEF    QUAN1C, SETBIT, LSW8, $SGQ1C
201 002266                                                   CMDDEF    NOQUAN1C, CLRBIT, LSW8, $SGQ1C
202 002276                                                   CMDDEF    QUAN2, SETBIT, LSW8, $SGQ2
203 002306                                                   CMDDEF    NOQUAN2, CLRBIT, LSW8, $SGQ2
204 002316                                                   CMDDEF    QUAN3, SETBIT, LSW8, $SGQ3
205 002326                                                   CMDDEF    NOQUAN3, CLRBIT, LSW8, $SGQ3
206 002336                                                   CMDDEF    OFF, CLRBIT, LSW8, $SGALL
207 002346                                                   TBLEND
208                                                  ;
209                                                  ;  Define options for the SET LOG command
210                                                  ;
211 000000                                                   TBLDEF    STLG, 1
212 002352                                                   CMDDEF    A*LL, STLGAL
213 002356                                                   CMDDEF    CLE*AN, STLGCN
214 002362                                                   CMDDEF    CLO*SED, STLGCL
215 002366                                                   CMDDEF    FI*LE, STLGFL
216 002372                                                   CMDDEF    I*NPUT, STLGIN
217 002376                                                   CMDDEF    O*UTPUT, STLGOT
218 002402                                                   CMDDEF    WR*ITE, STLGWR
219 002406                                                   CMDDEF    NOWR*ITE, STLGNW
220 002412                                                   TBLEND
221                                                  ;
222                                                  ;  Define options for the SET LOGOFF command
223                                                  ;
224 000000                                                   TBLDEF    STLO, 1
225 002416                                                   CMDDEF    FI*LE, STLOFL
226 002422                                                   TBLEND
227                                                  ;
228                                                  ;  Define options for the SET SUBPROCESS command
```

```
229                                    ;
230  000000                                 TBLDEF   SBP,1
231  002426                                 CMDDEF   FI*LE,SBPFIL
232  002432                                 TBLEND
233                                    ;
234                                    ;  Define options for the SET CTRLD command
235                                    ;
236  000000                                 TBLDEF   CTD,1
237  002436                                 CMDDEF   D*EBUG,STCDON
238  002442                                 CMDDEF   NOD*EBUG,STCDOF
239  002446                                 TBLEND
```

```
    1                                            ;--------------------------------------------------------------------
    2                                            ;   Data areas
    3                                            ;
    4 000000                                     HANSAV: .BLKW    5                    ;Space for .SAVESTATUS
    5 000012    012276                           R50CLO: .RAD50   /CLO/
    6 000014    013666                           R50C10: .RAD50   /C10/
    7 000016    000000                           WINFLG: .WORD    0
    8                                            ;
    9                                            ;   Flags stored in WINFLG
   10                                            ;
   11           000001                           WFDEL   =        1                    ;Delete the window
   12           000002                           WFREV   =        2                    ;Put terminal in reverse video
   13           000004                           WFNORM  =        4                    ;Put terminal in normal video mode
   14           000010                           WF132   =        10                   ;Put terminal in 132 column mode
   15           000020                           WF80    =        20                   ;Put terminal in 80 column mode
   16                                            ;
   17                                            ;   Terminal control sequences
   18                                            ;
   19 000020    033    133    077               TCREV:  .ASCII   <33>/[?5h/<200> ;Select reverse video display
      000023    065    150    200
   20 000026    033    133    077               TCNORM: .ASCII   <33>/[?5l/<200> ;Select normal video display
      000031    065    154    200
   21 000034    033    133    077               TC132:  .ASCII   <33>/[?3h/<200> ;Select 132 column mode
      000037    063    150    200
   22 000042    033    133    077               TC80:   .ASCII   <33>/[?3l/<200> ;Select 80 column mode
      000045    063    154    200
   23                                                    .EVEN
   24                                            ;
   25                                            ;   Emt to set line speed
   26                                            ;
   27 000050    000    154                       LSPEMT: .BYTE    0,154
   28 000052    000000                                   .WORD    0                    ;Line number
   29 000054    000000                                   .WORD    0                    ;Speed code
   30                                            ;
   31                                            ;   Emt to reinitialize data caching
   32                                            ;
   33 000056    000    126                       DCHNEW: .BYTE    0,126
   34 000060    000006                                   .WORD    6
   35                                            ;
   36                                            ;   Emt to set job priority
   37                                            ;
   38 000062    000    150                       PRIEMT: .BYTE    0,150
   39 000064    000000                                   .WORD    0
   40                                            ;
   41                                            ;   Emt to create window # 1
   42                                            ;
   43 000066    000    161                       WINMAK: .BYTE    0,161
   44 000070    001    001                               .BYTE    1,1
   45 000072    120    020                               .BYTE    80.,16.
   46 000074    000000 000000                            .WORD    0,0
   47                                            ;
   48                                            ;   Emt to map to window # 1
   49                                            ;
   50 000100    001    161                       WINMAP: .BYTE    1,161
   51 000102    001    000                               .BYTE    1,0
   52                                            ;
   53                                            ;   Emt to delete window # 1
```

```
 54                                              ;
 55 000104      002       161           WINDEL: .BYTE   2,161
 56 000106      001       000                   .BYTE   1,0
 57                                              ;
 58                                              ;   Emt to dismount all LD's
 59                                              ;
 60 000110      005       135           DMTALD: .BYTE   5,135
 61 000112   000000                             .WORD   0
```

```
    1                                              .SBTTL   SET command
    2                                      ;----------------------------------------------------------------------
    3                                      ;   Process the "SET" command.
    4                                      ;
    5 000114  004767  000000G      CMDSET: CALL     CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACES
    6 000120  004767  000000G              CALL     SKPSPC          ;Skip up to start of device name
    7 000124  010305                       MOV      R3,R5           ;SAVE COMMAND STRING POINTER
    8 000126  105067  000000G              CLRB     NOFLAG          ;ASSUME "NO" OPTION NOT SEEN YET
    9
   10                                      ;   If device name ends with a colon (e.g., "TT:") replace the colon
   11                                      ;   with a space.
   12                                      ;
   13 000132  112300               7$:     MOVB     (R3)+,RO        ;Get next char of device name
   14 000134  120027  000040               CMPB     RO,#40          ;Reached end of name?
   15 000140  101410                       BLOS     8$              ;Br if yes
   16 000142  120027  000057               CMPB     RO,#'/          ;Start of qualifiers?
   17 000146  001405                       BEQ      8$              ;Br if yes
   18 000150  120027  000072               CMPB     RO,#':          ;Is this character colon?
   19 000154  001366                       BNE      7$              ;Br if not
   20 000156  112743  000040               MOVB     #40,-(R3)       ;Replace colon with space
   21 000162  010503               8$:     MOV      R5,R3           ;Get back pointer to start of device name
   22                                      ;
   23                                      ;   Look up device name and see if we recognize it as a pseudo-device
   24                                      ;   such as TT, WILDCARD, etc.
   25                                      ;
   26 000164  012704  000000'              MOV      #SETHD,R4       ;GET TABLE OF PSEUDO-DEVICE NAMES
   27 000170  004767  000000G              CALL     SEARCH          ;SEARCH FOR DEVICE NAME
   28 000174  103401                       BCS      1$              ;BR IF NOT A PSEUDO-DEVICE NAME
   29
   30                                      ;   This is a pseudo-device name.
   31                                      ;   Jump off to appropriate processing routine.
   32                                      ;
   33 000176  000134                       JMP      @(R4)+          ;JUMP OFF TO PROCESSING ROUTINE
   34                                      ;
   35                                      ;   Device name is not recognized as a pseudo device.
   36                                      ;   Apply any logical device assignment to the device name.
   37                                      ;
   38 000200  010503               1$:     MOV      R5,R3           ;POINT BACK TO DEVICE NAME
   39 000202  005067  000002G              CLR      R50BUF+2        ;CLEAR 2ND WORD OF RAD50 NAME
   40 000206  004767  000000G              CALL     GTRD50          ;ACCRUE RAD50 VALUE
   41 000212  016700  000000G              MOV      R50BUF,RO       ;GET DEVICE NAME
   42 000216  005767  000002G              TST      R50BUF+2        ;MAKE SURE NAME ONLY 3 CHARS LONG
   43 000222  001404                       BEQ      2$              ;BR IF OK
   44 000224                               FABORT   #CSIMS2         ;INVALID DEVICE FOR SET
   45 000234  004767  000000G      2$:     CALL     ASNSRC          ;Search for device name in assign table
   46 000240  103404                       BCS      6$              ;Br if name was not assigned
   47 000242  016200  000000G              MOV      AT$DEV(R2),RO   ;Get the physical device name
   48 000246  010067  000000G              MOV      RO,R50BUF       ;Set new device name for SET
   49                                      ;
   50                                      ;   See if this is a SET LDn command
   51                                      ;
   52 000252  020067  000000G      6$:     CMP      RO,R50LD        ;SEE IF IT IS LD
   53 000256  103410                       BLO      3$              ;BR IF NOT
   54 000260  020067  000000G              CMP      RO,R50LD7       ;CHECK HIGH RANGE
   55 000264  101005                       BHI      3$              ;BR IF NOT AN LD UNIT
   56 000266  105767  000000G              TSTB     VLDSYS          ;IS LD SUPPORT GENNED IN?
   57 000272  001416                       BEQ      5$              ;BR IF NOT, maybe using an actual LD.TSX
```

```
    58 000274  000167  005560                        JMP     SETLD           ;GO PROCESS SET LD COMMAND
    59                                        ;
    60                                        ;   See if this is a SET CLn command
    61                                        ;   Note, that CL unit allocation is now checked only for some SET commands.
    62                                        ;
    63 000300  005727  000000G       3$:      TST     #CLTOTL         ;Are there any CL lines?
    64 000304  001411                          BEQ     5$              ;Br if not
    65 000306  004767  000000G                 CALL    CHKCLU          ;See if this is a CL or C1 unit (unit # to R0)
    66 000312  103003                          BCC     4$              ;BR if it is a valid CL or C1 unit
    67 000314  016700  000000G                 MOV     R5OBUF,R0       ;If not, we need to get back RAD50 device name
    68 000320  000403                          BR      5$              ;Try it as a "real" device
    69 000322  010002               4$:        MOV     R0,R2           ;Pass CL unit number in R2
    70 000324  000167  000000G                 JMP     SETCL           ;Process SET CL command
    71                                        ;
    72                                        ;   See if this device is allocated to another user before trying
    73                                        ;   to perform handler set code.
    74                                        ;
    75 000330  004767  000000G       5$:       CALL    CHKALC          ;Check for allocation conflict
    76 000334  000167  000470                  JMP     SETDEV
```

```
    1                                              ;-------------------------------------------------------------------
    2                                              ;  Process SET commands that have sub-options
    3                                              ;
    4                                              ;  Inputs:
    5                                              ;   R3 = Pointer to start of qualifier in command line.
    6                                              ;   R4 = Points to start of command table built with TBLDEF...TBLEND macros.
    7                                              ;
    8                                              ;  Outputs:
    9                                              ;   R3 = Pointer beyond end of qualifier.
   10                                              ;   R4 = Pointer to argument value word in command table.
   11                                              ;
   12                                              ;  See if set option is preceded by "NO".
   13                                              ;
   14 000340  004767  000000G       SETPRV: CALL    CKSYPV          ;Must have SYSPRV privilege
   15 000344  010246               SETSUB: MOV     R2,-(SP)
   16 000346  010546                       MOV     R5,-(SP)
   17 000350  122327  000072               CMPB    (R3)+,#':       ;WAS DEVICE NAME FOLLOWED BY COLON?
   18 000354  001401                       BEQ     3$              ;BR IF YES
   19 000356  005303                       DEC     R3              ;RESET POINTER IF NOT
   20 000360  004767  000000G       3$:    CALL    SKPSPC          ;SKIP OVER SPACES
   21 000364  111300                       MOVB    (R3),R0         ;Get next character from command
   22 000366  120027  000075               CMPB    R0,#'=          ;Is there an equal sign?
   23 000372  001403                       BEQ     5$              ;Br if yes -- skip over equal sign
   24 000374  120027  000072               CMPB    R0,#':          ;Colon separator?
   25 000400  001001                       BNE     6$              ;Br if not
   26 000402  005203               5$:     INC     R3              ;Skip over separator
   27 000404  010302               6$:     MOV     R3,R2           ;SAVE COMMAND STRING POINTER
   28 000406  004767  000000G               CALL    GTRD50          ;ACCRUE NEXT WORD IN RAD50 FORM
   29 000412  026767  000000G 000000G       CMP     R50BUF,R50NO    ;IS WORD "NO"?
   30 000420  001005                       BNE     1$              ;BR IF NOT
   31                                      ;
   32                                      ;  Option is preceeded by NO.  Concatenate NO with following qualifier.
   33                                      ;
   34 000422  010305                       MOV     R3,R5           ;GET POINTER INTO COMMAND PAST "NO"
   35 000424  004767  000000G               CALL    SKPSPC          ;SKIP OVER SPACES FOLLOWING "NO"
   36 000430  112325               4$:     MOVB    (R3)+,(R5)+     ;CONCATENATE OPTION WORD WITH "NO"
   37 000432  001376                       BNE     4$              ;MOVE ALL OF COMMAND
   38 000434  010203               1$:     MOV     R2,R3           ;GET BACK POINTER TO OPTION WORD
   39                                      ;
   40                                      ;  Look up the option word.
   41                                      ;
   42 000436  011404                       MOV     @R4,R4          ;GET POINTER TO TABLE OF OPTIONS FOR COMMAND
   43 000440  004767  000000G               CALL    SEARCH          ;LOOK FOR THE CORRECT OPTION WORD
   44 000444  103403                       BCS     BDSO            ;BR IF INVALID OPTION
   45                                      ;
   46                                      ;  Enter Option processing routine.
   47                                      ;
   48 000446  012605                       MOV     (SP)+,R5
   49 000450  012602                       MOV     (SP)+,R2
   50 000452  000134                       JMP     @(R4)+          ;PERFORM PROCESSING FOR OPTION
   51                                      ;
   52                                      ;  Invalid option.
   53                                      ;
   54 000454  005704               BDSO:   TST     R4              ;INVALID OR AMBIGUOUS OPTION?
   55 000456  001407                       BEQ     INVSOP          ;BR IF INVALID
   56 000460                               FABORT  #AMBOPT         ;AMBIGUOUS OPTION
   57 000470               INVLDO: .PURGE  #HANCHN         ;MAKE SURE THE CHANNEL IS CLOSED
```

        58 000476   INVSOP: FABORT #INVOPT   ; INVALID OPTION

```
     1                                            .SBTTL .          System parameters
     2                                   ;-----------------------------------------------------------------------
     3                                   ;   Set a system parameter value.
     4                                   ;
     5                                   ;   Inputs:
     6                                   ;    R4 Points to cell with address of parameter cell to be modified,
     7                                   ;     following word contains maximum legal value for parameter.
     8                                   ;
     9 000506   012746   177777         SETSVB:  MOV      #-1,-(SP)        ;Signal byte value
    10 000512   000401                           BR       SETSVC          ;Br to common code
    11 000514   005046                  SETSVL:  CLR      -(SP)           ;Signal word value
    12 000516   004767   000000G        SETSVC:  CALL     CKSYPV          ;Must have SYSPRV privilege
    13                                   ;
    14                                   ;   Accrue numeric value
    15                                   ;
    16 000522   004767   000000G                 CALL     ACRDEC          ;ACCRUE A DECIMAL VALUE
    17 000526   016400   000002                  MOV      2(R4),R0        ;GET MAX LEGAL VALUE FOR PARAMETER
    18                                   ;
    19                                   ;   Make sure the value is valid.
    20                                   ;   R1 = Accrued value,   R0 = Maximum legal value.
    21                                   ;
    22 000532   005701                  SETSVV:  TST      R1              ;CHECK FOR NEGATIVE VALUE SPECIFIED
    23 000534   002402                           BLT      3$              ;BR IF INVALID VALUE SPECIFIED
    24 000536   020100                           CMP      R1,R0           ;IS SPECIFIED VALUE OK?
    25 000540   101413                           BLOS     1$              ;BR IF OK
    26 000542   010001                  3$:      MOV      R0,R1           ;GET MAX LEGAL VALUE
    27 000544   010005                           MOV      R0,R5
    28 000546                                     .PRINT   #MAXAVL         ;VALUE TOO LARGE
    29 000554   004767   000000G                 CALL     PRTDEC          ;PRINT MAX LEGAL VALUE
    30 000560                                     .PRINT   #CRLF           ;END LINE
    31 000566   000406                           BR       4$              ;FINISHED
    32                                   ;
    33                                   ;   Store value into parameter cell
    34                                   ;
    35 000570   005726                  1$:      TST      (SP)+           ;Word or byte value?
    36 000572   001403                           BEQ      2$              ;Br if word
    37 000574   110174   000000                  MOVB     R1,@(R4)        ;Store byte value
    38 000600   000401                           BR       4$              ;Finish
    39 000602   010134                  2$:      MOV      R1,@(R4)+       ;STORE NEW VALUE INTO PARAMETER CELL
    40 000604   000167   000000G        4$:      JMP      RDCMD           ;FINISHED
    41                                   ;
    42                                   ;   SET NUMDC value
    43                                   ;
    44 000610   004767   000000G        SETNDC:  CALL     CKSYPV          ;Must have SYSPRV privilege
    45                                   ;
    46                                   ;   Accrue numeric value
    47                                   ;
    48 000614   004767   000000G                 CALL     ACRDEC          ;ACCRUE A DECIMAL VALUE
    49 000620   016700   000000G                 MOV      VNUMDC,R0       ;GET MAX LEGAL VALUE FOR PARAMETER
    50 000624   000742                           BR       SETSVV          ;Check and store value
```

```
     1                                           ;
     2                                           ;   SET CACHE = value
     3                                           ;
     4 000626   004767   000000G                STCHNB: CALL     CKSYPV              ;Must have SYSPRV privilege
     5                                           ;
     6                                           ;   Accrue numeric value
     7                                           ;
     8 000632   004767   000000G                        CALL     ACRDEC             ;ACCRUE A DECIMAL VALUE
     9 000636   016700   000000G                        MOV      CSHALC, RO         ;GET MAX LEGAL VALUE FOR PARAMETER
    10 000642   005701                                  TST      R1                 ;CHECK FOR NEGATIVE VALUE SPECIFIED
    11 000644   002402                                  BLT      3$                 ;BR IF INVALID VALUE SPECIFIED
    12 000646   020100                                  CMP      R1, RO             ;IS SPECIFIED VALUE OK?
    13 000650   101412                                  BLOS     1$                 ;BR IF OK
    14 000652   010001                          3$:     MOV      RO, R1             ;GET MAX LEGAL VALUE
    15 000654   010005                                  MOV      RO, R5
    16 000656                                           . PRINT  #MAXAVL            ;VALUE TOO LARGE
    17 000664   004767   000000G                        CALL     PRTDEC             ;PRINT MAX LEGAL VALUE
    18 000670                                           . PRINT  #CRLF              ;END LINE
    19                                           ;
    20                                           ;   Store value into parameter cell
    21                                           ;
    22 000676   010167   000000G                1$:     MOV      R1, VCSHNB         ;STORE NEW VALUE INTO PARAMETER CELL
    23                                           ;
    24                                           ;   Now clean out the data cache
    25                                           ;
    26 000702   012700   000056'                        MOV      #DCHNEW, RO        ;EMT to clean out the cache
    27 000706   104375                                  EMT      375
    28 000710   000167   000000G                        JMP      RDCMD              ;FINISHED
    29                                           ;
    30                                           ;   SET CTRLD [NO]DEBUG
    31                                           ;
    32 000714   105767   000000G                STCDON: TSTB     VDBFLG             ;Was debugger genned into the system?
    33 000720   001004                                  BNE      1$                 ;Br if yes
    34 000722                                           FABORT   #EM$NPD            ;No program debugger
    35 000732   032767   000000G 000000G 1$:            BIT      #PO$DBG, PRIVCO    ;Can we use debugger?
    36 000740   001004                                  BNE      2$                 ;Br if yes
    37 000742                                           FABORT   #EM$NAD            ;Not allowed to use debugger
    38 000752   052761   000000G 000000G 2$:            BIS      #$CTRLD, LSW9(R1)  ;Enable ctrl-D debugger entry
    39 000760   000167   000000G                        JMP      RDCMD
    40 000764   042761   000000G 000000G STCDOF: BIC             #$CTRLD, LSW9(R1)  ;Disable ctrl-D debugger entry
    41 000772   000167   000000G                        JMP      RDCMD
    42                                           ;
    43                                           ;   SET EMT TRACE
    44                                           ;
    45 000776   032767   000000G 000000G SETEMT: BIT             #PO$DBG, PRIVCO    ;Are we allowed to use debugger?
    46 001004   001004                                  BNE      1$                 ;Br if yes
    47 001006                                           FABORT   #EM$NAD            ;Not allowed to use debugger
    48 001016   052761   000000G 000000G 1$:            BIS      #$EMTTR, LSW6(R1)  ;Enable EMT trace
    49 001024   000167   000000G                        JMP      RDCMD
```

```
   1                                                .SBTTL .          Real device
   2                                      ;----------------------------------------------------------------------
   3                                      ;  The device name is not a pseudo-device.
   4                                      ;  See if it is the name of a real device.
   5                                      ;
   6 001030  016701  000000G     SETDEV:  MOV     R50BUF,R1        ;GET THE DEVICE NAME AND UNIT
   7 001034  005000               CLR     R0              ;CLEAR HIGH ORDER REGISTER
   8 001036  071027  000050       DIV     #50,R0          ;STRIP THE LAST RAD 50 CHARACTER
   9 001044  010146               MOV     R1,-(SP)        ;SAVE THE UNIT NUMBER
  10 001044  070027  000050       MUL     #50,R0          ;GET THE DEVICE NAME SHIFTED
  11 001050  010167  000000G      MOV     R1,R50BUF       ;AND STORE BACK INTO THE RAD 50 BUFFER
  12 001054  012601               MOV     (SP)+,R1        ;GET THE UNIT NUMBER
  13 001056  001407               BEQ     2$              ;BR IF NO UNIT NUMBER WAS SPECIFIED
  14 001060  162701  000036       SUB     #36,R1          ;NORMALIZE TO ZERO
  15 001064  002450               BLT     SETIVD          ;ERROR IF LESS THAN ZERO
  16 001066  020127  000007       CMP     R1,#7           ;CHECK UNIT RANGE
  17 001072  003045               BGT     SETIVD          ;ERROR IF GREATER THAN SEVEN
  18 001076  000402               BR      3$              ;CONTINUE
  19 001076  012701  100000  2$:  MOV     #100000,R1      ;SET NO UNIT WAS SPECIFIED FLAG
  20 001102  010167  000000G 3$:  MOV     R1,DEVUNT       ;SAVE THE UNIT SPECIFIED
  21 001106  122327  000072       CMPB    (R3)+,#':       ;WAS DEVICE NAME FOLLOWED BY COLON?
  22 001112  001401               BEQ     4$              ;BR IF YES
  23 001114  005303               DEC     R3              ;BACKUP POINTER
  24                                      ;
  25                                      ;  Lookup the handler file and read in the first 2 blocks.
  26                                      ;
  27 001116  016767  000000G 000002G 4$:  MOV     R50BUF,HNBUF+2  ;SET UP HANDLER NAME
  28 001124                       .LOOKUP #XAREA,#HANCHN,#HNBUF;TRY TO LOOKUP HANDLER FILE
  29 001144  103420               BCS     SETIVD          ;BR IF CAN'T FIND HANDLER
  30 001146                       .READW  #XAREA,#HANCHN,#BLKO,#512.,#0 ;READ IN 1ST 2 BLOCKS
  31 001204  103007               BCC     SETCKP          ;Branch if read ok
  32                                      ;
  33                                      ;  Invalid device name.
  34                                      ;
  35 001206              SETIVD:  .PURGE  #HANCHN         ;MAKE SURE CHANNEL IS CLOSED
  36 001214                       FABORT  #CSIMS2         ;INVALID DEVICE NAME
  37                                      ;
  38                                      ;  We are performing a set to a device handler.
  39                                      ;  User must be privileged to do this.
  40                                      ;
  41 001224              SETCKP:  .SAVEST #XAREA,#HANCHN,#HANSAV ;Save channel status in HANSAV
  42 001244  032767  000000G 000000G      BIT     #PO$OPR,PRIVCO  ;Is user privileged?
  43 001252  001007               BNE     NXTOPT          ;Br if yes
  44 001254                       .PURGE  #HANCHN         ;Close handler channel
  45 001262                       FABORT  #EM$OPR         ;Not-privileged error message
```

```
     1                                          ;   Accrue the set option word.
     2                                          ;   Accrue the set option word.
     3                                          ;
     4 001272  105067  000000G        NXTOPT:  CLRB    NOFLAG           ;RESET THE "NO" OPTION FLAG
     5 001276  004767  000000G                 CALL    SKPSPC           ;SKIP OVER ANY SPACES
     6                                          ;   Check for "NO" preceding the option word.
     7 001302  010302                           MOV     R3,R2            ;SAVE COMMAND STRING POINTER
     8 001304  004767  000000G                 CALL    GETKCH           ;GET NEXT COMMAND CHARACTER
     9 001310  120027  000116                  CMPB    RO,#'N           ;IS IT "N"?
    10 001314  001012                           BNE     7$               ;BR IF NOT
    11 001316  004767  000000G                 CALL    GETKCH           ;CHECK NEXT CHARACTER
    12 001322  120027  000117                  CMPB    RO,#'O           ;IS IT "O"?
    13 001326  001005                           BNE     7$               ;BR IF NOT
    14 001330  105267  000000G                 INCB    NOFLAG           ;REMEMBER "NO" SPECIFIED
    15 001334  004767  000000G                 CALL    SKPSPC           ;SKIP OVER SPACES FOLLOWING "NO"
    16 001340  000401                           BR      9$
    17 001342  010203                  7$:      MOV     R2,R3            ;RESET POINTER TO START OF OPTION WORD
    18                                          ;   Accrue the option word.
    19 001344  004767  000000G         9$:      CALL    GTRD50           ;ACCRUE AS A RAD50 VALUE
    20                                          ;
    21                                          ;   Lookup the option word in the handler header.
    22                                          ;
    23 001350  012704  000400G        SETOPT:  MOV     #BLKO+400,R4     ;POINT TO START OF HANDLER TABLE OF SET OPTIONS
    24 001354  005764  000000G         10$:     TST     SH$VAL(R4)       ;HAVE WE REACHED THE END OF THE TABLE?
    25 001360  001002                           BNE     11$              ;CONTINUE CHECKING
    26 001362  000167  177102                  JMP     INVLDO           ;BR IF YES -- CAN'T FIND OPTION WORD
    27 001366  026764  000000G 000000G 11$:     CMP     R5OBUF,SH$NAM(R4);COMPARE NAME WITH THIS ENTRY
    28 001374  001004                           BNE     2$               ;BR IF MISMATCH
    29 001404  026764  000002G 000002G         CMP     R5OBUF+2,SH$NAM+2(R4)
    30 001404  001403                           BEQ     9$               ;BR IF FOUND ENTRY FOR THIS OPTION
    31 001406  062704  000000G         2$:      ADD     #SH$$SZ,R4       ;POINT TO NEXT OPTION ENTRY
    32 001412  000760                           BR      10$              ;CONTINUE SEARCH
    33                                          ;
    34                                          ;   Found entry for this option.
    35                                          ;
    36 001414  116402  000000G         9$:      MOVB    SH$RTN(R4),R2    ;GET OFFSET TO ROUTINE TO CALL
    37 001420  042702  177400                  BIC     #^C377,R2        ;CLEAR SIGN EXTENSION
    38 001424  006302                           ASL     R2               ;CONVERT FROM WORD TO BYTE OFFSET
    39 001426  062702  000400G                 ADD     #BLKO+400,R2     ;ADD BASE ADDRESS
    40                                          ;   See if "NO" option is allowed.
    41 001432  105767  000000G                 TSTB    NOFLAG           ;WAS "NO" SPECIFIED WITH OPTION?
    42 001436  001410                           BEQ     3$               ;BR IF NOT
    43 001440  132764  000000G 000000G         BITB    #SO$NO,SH$FLG(R4);IS NO ALLOWED WITH THIS OPTION?
    44 001446  001002                           BNE     14$              ;BR IF YES
    45 001450  000167  000464                  JMP     SETIVS           ;INVALID OPTION
    46 001454  062702  000004          14$:     ADD     #4,R2            ;GET ADDRESS OF SET ROUTINE FOR NO-OPTION
    47                                          ;   See if option takes a numeric parameter.
    48 001460  132764  000000G 000000G 3$:      BITB    #SO$NVL,SH$FLG(R4);DOES OPTION REQUIRE A NUMERIC PARAMETER?
    49 001466  001433                           BEQ     4$               ;BR IF NOT
    50 001470  004767  000000G                 CALL    SKPSPC           ;SKIP OVER ANY SPACES
    51 001474  122327  000075                  CMPB    (R3)+,#'=        ;IS OPTION WORD FOLLOWED BY EQUAL SIGN?
    52 001500  001401                           BEQ     7$               ;BR IF YES
    53 001502  005303                           DEC     R3               ;BACKUP POINTER
    54 001504  004767  000000G         7$:      CALL    SKPSPC           ;SKIP OVER ANY SPACES
    55 001510  010305                           MOV     R3,R5            ;SAVE POINTER TO START OF NUMBER
    56 001512  132764  000000G 000000G         BITB    #SO$OCT,SH$FLG(R4);IS NUMERIC VALUE OCTAL?
    57 001520  001003                           BNE     6$               ;BR IF YES
```

```
 58 001522  004767  000000G                CALL    ACRDEC          ;ACCRUE A DECIMAL VALUE
 59 001526  000402                         BR      5$
 60 001530  004767  000000G        6$:     CALL    ACROCT          ;ACCRUE AN OCTAL VALUE
 61 001534  010100                 5$:     MOV     R1,R0           ;GET VALUE IN R0 FOR SET ROUTINE
 62 001536  020305                         CMP     R3,R5           ;DID WE ACTUALLY GET A NUMBER?
 63 001540  001002                         BNE     15$             ;BR IF YES
 64 001542  000167  000372                 JMP     SETIVS          ;BR IF NOT
 65 001546  122327  000056        15$:     CMPB    (R3)+,#'.       ;WAS DECIMAL POINT SPECIFIED?
 66 001552  001401                         BEQ     4$              ;BR IF YES
 67 001554  005303                         DEC     R3              ;BACKUP POINTER
 68                                 ;
 69                                 ;  Open channel 17 (which is normally used for KMON overlays) to
 70                                 ;  the handler file so that some handlers can use this channel to
 71                                 ;  read in and updated additional blocks in the handler.
 72                                 ;
 73 001556  010046                 4$:     MOV     R0,-(SP)        ;Save numeric parameter value
 74 001560                                 .PURGE  #17             ;Purge channel 17 (KMON overlay channel)
 75 001566                                 .REOPEN #XAREA,#17,#HANSAV ;Open channel 17 to handler file
 76 001606  012600                         MOV     (SP)+,R0        ;Restore numeric parameter value
 77                                 ;
 78                                 ;  Call routine in handler that actually does the set.
 79                                 ;  On entry the following registers are set up:
 80                                 ;   R0 = Optional numeric parameter value.
 81                                 ;   R3 = Fixed option value found in option word (SH$VAL)
 82                                 ;
 83 001610  010346                         MOV     R3,-(SP)        ;SAVE COMMAND STRING POINTER
 84 001612  016403  000000G                MOV     SH$VAL(R4),R3   ;GET VALUE FROM OPTION ENTRY
 85 001616  016701  000000G                MOV     DEVUNT,R1       ;GET THE DEVICE UNIT SPECIFIED
 86 001622  000241                         CLC                     ;CLEAR C-BIT BEFORE ENTERING HANDLER
 87 001624  004712                         CALL    @R2             ;CALL SET PROCESSING ROUTINE IN HANDLER
 88 001626  012603                         MOV     (SP)+,R3        ;RECOVER COMMAND STRING POINTER
 89 001630  012702  000000                 MOV     #0,R2           ;SAY NO ERROR (DON'T AFFECT C-FLAG)
 90 001634  103001                         BCC     13$             ;BR IF NO ERROR OCCURRED
 91 001636  005202                         INC     R2              ;SET ERROR FLAG IN R2
 92                                 ;
 93                                 ;  Restore status of HANCHN and channel 17
 94                                 ;
 95 001640                         13$:    .PURGE  #17             ;Purge channel 17
 96 001646                                 .REOPEN #XAREA,#17,#KMNCHN ;Reopen channel 17 to KMON
 97 001666                                 .REOPEN #XAREA,#HANCHN,#HANSAV ;Reopen HANCHN to handler
 98 001706  005702                         TST     R2              ;Any error detected in handler?
 99 001710  001113                         BNE     SETIVS          ;Br if yes
100                                 ;
101                                 ;  See if there are more options.
102                                 ;
103 001712  004767  000000G                CALL    SKPSPC          ;SKIP SPACES
104 001716  112300                         MOVB    (R3)+,R0        ;GET NEXT CHAR
105 001720  001406                         BEQ     SETUP           ;BR IF REACHED END
106 001722  120027  000054                 CMPB    R0,#',          ;IS COMMA THE DELIMITER?
107 001726  001401                         BEQ     12$             ;BR IF YES
108 001730  005303                         DEC     R3              ;BACKUP STRING POINTER
109 001732  000167  177334        12$:     JMP     NXTOPT          ;GO PROCESS THE NEXT OPTION
```

```
   1                                        ;
   2                                        ; Finished with all options.
   3                                        ; Write updated handler back to its disk file.
   4                                        ;
   5 001736                      SETUP:      .WRITW  #XAREA,#HANCHN,#BLKO,#512.,#0
   6 001774                                  .CLOSE  #HANCHN
   7                                        ;
   8                                        ; Now try to update the running handler in memory.
   9                                        ;
  10 002002  016702  000000G                MOV     NUMDEV,R2          ;GET INDEX TO LAST DEVICE IN DEVICE TABLES
  11 002006  016700  000002G                MOV     HNBUF+2,R0         ;Get device name
  12 002012  020062  000000G       1$:      CMP     R0,PNAME(R2)       ;LOOKUP DEVICE NAME
  13 002016  001412                         BEQ     3$                 ;BR IF FOUND
  14 002020  162702  000002                 SUB     #2,R2              ;MORE ENTRIES TO CHECK?
  15 002024  003372                         BGT     1$                 ;BR IF YES
  16 002026                                 FWARN   #EM$HNI            ;Warn that handler is not installed
  17 002042  000434                         BR      2$
  18                                        ;
  19                                        ; We have located the entry for the handler.
  20                                        ; See if we are allowed to change the in-memory handler image.
  21                                        ;
  22 002044  032762  000000G 000000G 3$:    BIT     #DX$NST,DVFLAG(R2) ;Are we allowed to reload running handler?
  23 002052  001407                         BEQ     4$                 ;Br if yes
  24 002054                                 FWARN   #TM$MRS            ;Tell user that he must reboot the system
  25 002070  000421                         BR      2$
  26 002072  012700  000000G       4$:      MOV     #HUPARG,R0         ;POINT TO ARG BLOCK FOR UPDATE INFO
  27 002076  010260  000004                 MOV     R2,4(R0)           ;SET DEVICE INDEX NUMBER
  28 002102  005067  001006G                CLR     BLKO+1006          ;Clear LQE in handler image
  29 002106  005067  001010G                CLR     BLKO+1010          ;Clear CQE in handler image
  30 002112  012760  001000G 000006         MOV     #BLKO+1000,6(R0)   ;SET ADDRESS OF START OF NEW HANDLER CODE
  31 002120  104375                         EMT     375                ;TRY TO UPDATE RUNNING HANDLER
  32 002122  103004                         BCC     2$                 ;BR IF SUCCESSFUL
  33 002124                                 FABORT  #HANBSY            ;HANDLER WAS ACTIVE
  34 002134  000167  000000G       2$:      JMP     RDCMD              ;FINISHED
  35                                        ;
  36                                        ; Invalid command
  37                                        ;
  38 002140                      SETIVS:     .PURGE  #HANCHN            ;MAKE SURE CHANNEL IS CLOSED
  39 002146                                 FABORT  #CSIMS1
```

```
    1                                          ;
    2                                          ;   Process options which only involve setting/resetting flags.
    3                                          ;
    4                                          ;   Set a bit in a user table
    5                                          ;
    6 002156  004767  000000G       PRVSET:  CALL    CKPRIV            ;REQUIRE PRIVILEGE
    7 002162  012405               SETBIT:  MOV     (R4)+,R5          ;POINT TO TABLE
    8 002164  060105                        ADD     R1,R5             ;POINT TO TABLE ENTRY FOR THIS USER
    9 002166  051415                        BIS     @R4,@R5           ;SET THE DESIRED FLAG
   10 002170  000167  000000G               JMP     RDCMD
   11                                        ;
   12                                        ;   Reset a bit in a user table
   13                                        ;
   14 002174  004767  000000G       PRVCLR:  CALL    CKPRIV            ;REQUIRE PRIVILEGE
   15 002200  012405               CLRBIT:  MOV     (R4)+,R5          ;POINT TO TABLE
   16 002202  060105                        ADD     R1,R5             ;POINT TO TABLE ENTRY FOR THIS USER
   17 002204  041415                        BIC     @R4,@R5           ;RESET THE DESIRED FLAG
   18 002206  000167  000000G               JMP     RDCMD
   19                                        ;
   20                                        ;   Routine to store a value into a word cell
   21                                        ;
   22 002212  012405               STRWRD:  MOV     (R4)+,R5          ;POINT TO WORD
   23 002214  011415                        MOV     (R4),(R5)         ;SET VALUE IN WORD
   24 002216  000167  000000G               JMP     RDCMD
   25                                        ;
   26                                        ;   Routine to accrue a byte value and store it into a line table.
   27                                        ;
   28 002222  122327  000075       SETBYT:  CMPB    (R3)+,#'=         ;EQUAL SIGN SHOULD BE NEXT CHAR
   29 002226  001014                        BNE     NOEQL             ;BR IF ERROR
   30 002230  122327  000040       2$:      CMPB    (R3)+,#'          ;SKIP INTERVENING SPACES
   31 002234  001775                        BEQ     2$
   32 002236  005303                        DEC     R3
   33 002240  004767  000000G               CALL    ACRDEC            ;ACCRUE THE DECIMAL VALUE
   34 002244  116702  000000G               MOVB    CORUSR,R2         ;GET LINE INDEX #
   35 002250  061402                        ADD     @R4,R2            ;POINT TO LINE ENTRY IN TABLE
   36 002252  110112                        MOVB    R1,@R2            ;STORE VALUE INTO TABLE
   37 002254  000167  000000G               JMP     RDCMD
   38                                        ;
   39 002260                        NOEQL:   FABORT  #MISSEQ           ;MISSING EQUAL SIGN
   40                                        ;
   41                                        ;   Set EDIT EDIT
   42                                        ;
   43 002270  042761  000000C 000000G SEEDIT: BIC    #$TECO!$KED,LSW5(R1)
   44 002276  000167  000000G               JMP     RDCMD
   45                                        ;
   46                                        ;   Set EDIT TECO
   47                                        ;
   48 002302  042761  000000G 000000G SETECO: BIC    #$KED,LSW5(R1)
   49 002310  052761  000000G 000000G        BIS    #$TECO,LSW5(R1)
   50 002316  000167  000000G               JMP     RDCMD
   51                                        ;
   52                                        ;   SET EDIT KED/K52
   53                                        ;
   54 002322                        SEK52:
   55 002322  042761  000000G 000000G SEKED: BIC    #$TECO,LSW5(R1)
   56 002330  052761  000000G 000000G        BIS    #$KED,LSW5(R1)
   57 002336  000167  000000G               JMP     RDCMD
```

```
 58                                                      ;
 59                                                      ;    SET UCL FIRST
 60                                                      ;
 61 002342  052761   000000G 000000G STUCLF:  BIS       #$UCLCF,LSW7(R1);Set UCL-first flag
 62 002350  042761   000000C 000000G          BIC       #<$UCLCM!$UCLCL>,LSW7(R1) ;Clear call-middle & call-last flags
 63 002356  000167   000000G              JMP       RDCMD
 64                                                      ;
 65                                                      ;    SET UCL MIDDLE
 66                                                      ;
 67 002362  052761   000000G 000000G STUCLM:  BIS       #$UCLCM,LSW7(R1);Set UCL-middle flag
 68 002370  042761   000000C 000000G          BIC       #<$UCLCF!$UCLCL>,LSW7(R1) ;Clear call-first & call-last flags
 69 002376  000167   000000G              JMP       RDCMD
 70                                                      ;
 71                                                      ;    SET UCL LAST
 72                                                      ;
 73 002402  052761   000000G 000000G STUCLL:  BIS       #$UCLCL,LSW7(R1);Set UCL-last flag
 74 002410  042761   000000C 000000G          BIC       #<$UCLCF!$UCLCM>,LSW7(R1) ;Clear call-first & call-middle flag
 75 002416  000167   000000G              JMP       RDCMD
 76                                                      ;
 77                                                      ;    SET UCL NONE
 78                                                      ;
 79 002422  042761   000000C 000000G STUCLN:  BIC       #<$UCLCF!$UCLCM!$UCLCL>,LSW7(R1) ;Clear all UCL-call flags
 80 002430  000167   000000G              JMP       RDCMD
 81                                                      ;
 82                                                      ;    SET KMON IND
 83                                                      ;
 84 002434  005767   000000G          SETIND:  TST       INDSAV            ; IS IND AVAILABLE?
 85 002440  001432                             BEQ       1$                ; BR IF NOT
 86 002442  052761   000000G 000000G          BIS       #$INDDF,LSW5(R1);SAY WE SHOULD USE IND
 87 002450                                      .GVAL     #XAREA,#<CFSTS-MONVEC>   ;GET CURRENT COMMAND FILE FLAGS
 88 002470  010002                             MOV       R0,R2
 89 002472  052702   000000G                   BIS       #CF$IND,R2                 ; SET IND FLAG
 90 002476                                      .PVAL     #XAREA,#<CFSTS-MONVEC>,R2;STORE UPDATED FLAGS
 91 002522  000167   000000G              JMP       RDCMD
 92 002526                             1$:    FABORT    #NOIND
 93                                                      ;
 94                                                      ;    SET KMON NO IND
 95                                                      ;
 96 002536  042761   000000G 000000G RSTIND:  BIC       #$INDDF,LSW5(R1);CLEAR IND FLAG
 97 002544                                      .GVAL     #XAREA,#<CFSTS-MONVEC>   ;GET CURRENT COMMAND FILE FLAGS
 98 002564  010002                             MOV       R0,R2
 99 002566  042702   000000G                   BIC       #CF$IND,R2                 ;RESET IND FLAG
100 002572                                      .PVAL     #XAREA,#<CFSTS-MONVEC>,R2;STORE UPDATED FLAGS
101 002616  000167   000000G              JMP       RDCMD
```

```
     1                                              .SBTTL  .          Prompt
     2                                      ;------------------------------------------------------------------
     3                                      ;   SET PROMPT string
     4                                      ;
     5 002622                              SETPRT:
     6                                      ;
     7                                      ;   Skip any leading spaces in front of prompt string
     8                                      ;
     9 002622  004767  000000G                     CALL    SKPSPC
    10 002626  012704  000000G                     MOV     #KMPRMT,R4      ;Point to prompt holder in context block
    11                                      ;
    12                                      ;   Determine if prompt string is enclosed in quotes
    13                                      ;
    14 002632  111300                              MOVB    (R3),R0         ;Get first char of prompt string
    15 002634  120027  000047                      CMPB    R0,#47          ;Is string enclosed in " ' " characters?
    16 002640  001403                              BEQ     2$              ;Br if yes
    17 002642  120027  000042                      CMPB    R0,#42          ;How about " " " ?
    18 002646  001007                              BNE     3$              ;Br if not
    19                                      ;
    20                                      ;   String is enclosed in quotes
    21                                      ;
    22 002650  004767  000000G             2$:      CALL    ACRSTR          ;Accrue the string
    23 002654  010002                              MOV     R0,R2           ;Get length of the string
    24 002656  001417                              BEQ     6$              ;Br if string is empty
    25 002660  012703  000000G                     MOV     #BLK0,R3        ;Point to buffer with string
    26 002664  000403                              BR      10$
    27                                      ;
    28                                      ;   String is not enclosed in quotes
    29                                      ;
    30 002666  020302                      3$:      CMP     R3,R2           ;Anything in prompt string?
    31 002670  103012                              BHIS    6$              ;Br if not
    32 002672  160302                              SUB     R3,R2           ;Determine length of prompt string
    33                                      ;
    34                                      ;   Make sure string is not too long
    35                                      ;
    36 002674  020227  000000G             10$:     CMP     R2,#MXPRMT      ;Is string too long?
    37 002700  101411                              BLOS    5$              ;Br if ok
    38 002702                                      FERR    #BADPMT         ;Prompt string too long
    39 002716  112724  000056             6$:      MOVB    #'.,(R4)+       ;Set prompt to "."
    40 002722  000402                              BR      8$
    41                                      ;
    42                                      ;   Set prompt string
    43                                      ;
    44 002724  112324                      5$:      MOVB    (R3)+,(R4)+     ;Move prompt string to KMON context block
    45 002726  077202                              SOB     R2,5$
    46 002730  112714  000200             8$:      MOVB    #200,(R4)       ;Terminate the prompt string
    47                                      ;
    48                                      ;   Finished
    49                                      ;
    50 002734  000167  000000G             9$:      JMP     RDCMD           ;Go get next command
```

```
     1                                                     .SBTTL   .          Syspassword
     2                                          ;--------------------------------------------------------------------
     3                                          ;   SET SYSPASSWORD command
     4                                          ;
     5 002740                                   SETSYP:
     6                                          ;
     7                                          ;   User must have SYSPRV privilege
     8                                          ;
     9 002740  004767  000000G                         CALL    CKSYPV              ;Make sure user has SYSPRV privilege
    10                                          ;
    11                                          ;   Accrue the password string
    12                                          ;
    13 002744  004767  000000G                         CALL    ACRTXT             ;Accrue the text string
    14                                          ;
    15                                          ;   Make sure it's not too long
    16                                          ;
    17 002750  020027  000024                         CMP     R0,#20.            ;Compare with max legal length
    18 002754  101404                                 BLOS    1$                 ;Br if ok
    19 002756                                         FABORT  #EM$SPL            ;Password string is too long
    20                                          ;
    21                                          ;   Move password to password buffer
    22                                          ;
    23 002766  012702  000000G                  1$:    MOV     #BLK0,R2           ;Point to accrued password
    24 002772  012703  000000G                         MOV     #SYPSWD,R3         ;Point to system buffer
    25 002776  112223                           2$:    MOVB    (R2)+,(R3)+        ;Move password string
    26 003000  001376                                  BNE     2$                 ;Loop if more to move
    27                                          ;
    28                                          ;   Finished
    29                                          ;
    30 003002  000167  000000G                         JMP     RDCMD
```

```
     1                                                      .SBTTL  .         Endstartup
     2                                             ;---------------------------------------------------------------------
     3                                             ;   SET ENDSTARTUP command
     4                                             ;   (Signal end of privileged portion of start-up command file)
     5                                             ;
     6 003006   042761   000000G 000000G SETESU:  BIC      #$NOIN,LSW3(R1) ;Allow input to be accepted for line
     7 003014   042761   000000G 000000G          BIC      #$SUCF,LSW9(R1) ;Say not in start-up command file
     8 003022   000167   000000G                  JMP      RDCMD
```

```
   1                                                   .SBTTL .        Window
   2                                          ;--------------------------------------------------------------
   3                                          ;   SET WINDOW command
   4                                          ;
   5 003026                                   SETWIN:
   6                                          ;
   7                                          ;   User must be authorized to create global regions in order to use windows
   8                                          ;
   9 003026  032767  000000G 000000G             BIT     #P2$CGR,PRIVC2   ;May job create global regions?
  10 003034  001004                              BNE     10$              ;Br if yes
  11 003036                                       FABORT  #EM$WCM          ;Not privileged to do this
  12                                          ;
  13                                          ;   Initialize some cells
  14                                          ;
  15 003046  112767  000120  175016  10$:      MOVB    #80.,WINMAK+4    ;Set window width to 80 columns
  16 003054  112767  000020  175011            MOVB    #16.,WINMAK+5    ;Set default max scroll lines
  17 003062  005067  174730                    CLR     WINFLG           ;No flags yet
  18                                          ;
  19                                          ;   Do command parsing
  20                                          ;
  21 003066  012704  001460'                   MOV     #WINHD,R4        ;Point to option driver list
  22 003072  004767  000000G                   CALL    SCNOPS           ;Process all command options
  23                                          ;
  24                                          ;   Now see which options were specified
  25                                          ;
  26 003076  016705  174714          12$:      MOV     WINFLG,R5        ;Pick up option flags
  27                                          ;
  28                                          ;   See if OFF was specified
  29                                          ;
  30 003102  032705  000001                    BIT     #WFDEL,R5        ;Was OFF option specified?
  31 003106  001404                             BEQ     1$               ;Br if not
  32 003110  012700  000104'                    MOV     #WINDEL,R0       ;Delete the window
  33 003114  104375                             EMT     375
  34 003116  000455                             BR      20$
  35                                          ;
  36                                          ;   If WIDE option was specified, set window width to 132 columns
  37                                          ;
  38 003120  032705  000010          1$:       BIT     #WF132,R5        ;Was WIDE option specified?
  39 003124  001403                             BEQ     2$               ;Br if not
  40 003126  112767  000204  174736             MOVB    #132.,WINMAK+4   ;Set window width to 132 columns
  41                                          ;
  42                                          ;   Try to make the window
  43                                          ;
  44 003134  032761  000000C 000000G 2$:       BIT     #<VT100!VT200!VT52>,LTRMTP(R1) ;VT100, VT200 or VT52
  45 003142  001004                             BNE     13$              ;Br if yes
  46 003144                                     FABORT  #EM$SLT          ;Invalid terminal type
  47 003154  012700  000066'          13$:      MOV     #WINMAK,R0       ;Point to window-make arg block
  48 003160  104375                             EMT     375              ;Try to create the window
  49 003162  103435                             BCS     30$              ;Br if error on window creation
  50                                          ;
  51                                          ;   Select the window
  52                                          ;
  53 003164  012700  000100'                    MOV     #WINMAP,R0       ;EMT to map to the window
  54 003170  104375                             EMT     375
  55                                          ;
  56                                          ;   If WIDE was specified, send string to put term in 132 col mode
  57                                          ;
```

```
    58 003172  032705  000010                    BIT     #WF132,R5       ;Was WIDE option specified?
    59 003176  001403                            BEQ     3$              ;Br if not
    60 003200                                     .PRINT  #TC132          ;Put terminal in 132 column mode
    61                                    ;
    62                                    ;   If NARROW was specified, send string to put term in 80 col mode
    63                                    ;
    64 003206  032705  000020            3$:      BIT     #WF80,R5        ;Was NARROW option specified?
    65 003212  001403                            BEQ     4$              ;Br if not
    66 003214                                     .PRINT  #TC80           ;Put terminal in 80 column mode
    67                                    ;
    68                                    ;   If REVERSE was specified, put terminal in reverse video mode
    69                                    ;
    70 003222  032705  000002            4$:      BIT     #WFREV,R5       ;Was REVERSE option specified?
    71 003226  001403                            BEQ     5$              ;Br if not
    72 003230                                     .PRINT  #TCREV          ;Put terminal in reverse video mode
    73                                    ;
    74                                    ;   If NORMAL was specified, put terminal in normal video mode
    75                                    ;
    76 003236  032705  000004            5$:      BIT     #WFNORM,R5      ;Was NORMAL option specified?
    77 003242  001403                            BEQ     20$             ;Br if not
    78 003244                                     .PRINT  #TCNORM         ;Put terminal in normal video mode
    79                                    ;
    80                                    ;   Finished
    81                                    ;
    82 003252  000167  0000000           20$:     JMP     RDCMD
    83                                    ;
    84                                    ;   Error occurred which making window
    85                                    ;
    86 003256  113702  0000000           30$:     MOVB    @#ERRLOC,R2     ;Get EMT error code
    87 003262  020227  000004                     CMP     R2,#MAXWEM      ;Is it too large?
    88 003266  103401                            BLO     31$             ;Br if not
    89 003270  005002                            CLR     R2              ;Use message for error 0 if too big
    90 003272  006302            31$:     ASL     R2              ;Get word table index
    91 003274  016202  003306'                    MOV     WEM(R2),R2      ;Get address of error message
    92 003300                                     FABORT  R2              ;Print the error message
    93                                    ;
    94                                    ;   Table of window-creation error messages
    95                                    ;
    96 003306  0000000           WEM:     .WORD   EM$WC0
    97 003310  0000000                    .WORD   EM$WC1
    98 003312  0000000                    .WORD   EM$WC2
    99 003314  0000000                    .WORD   EM$WC3
   100         000004           MAXWEM  =       <.-WEM>/2
```

```
    1                                               ;-------------------------------------------------------------------------
    2                                               ;   Process COLUMN=n option of SET WINDOW command
    3                                               ;
    4 003316  010146                                WINCOL: MOV     R1,-(SP)
    5                                               ;
    6                                               ;   Accrue the column parameter value
    7                                               ;
    8 003320  004767  000000G                               CALL    ACRDEC
    9                                               ;
   10                                               ;   Store into EMT argument block
   11                                               ;
   12 003324  110167  174542                                MOVB    R1,WINMAK+4      ;Set column width in EMT arg block
   13                                               ;
   14                                               ;   Finished
   15                                               ;
   16 003330  012601                                        MOV     (SP)+,R1
   17 003332  000207                                        RETURN
   18
   19                                               ;-------------------------------------------------------------------------
   20                                               ;   Process SCROLL[=n] option which specifies the maximum number
   21                                               ;   of scroll lines.
   22                                               ;
   23 003334  010146                                WINSCR: MOV     R1,-(SP)
   24                                               ;
   25                                               ;   If no parameter is specified, allow unlimited scrolling
   26                                               ;
   27 003336  004767  000000G                               CALL    SKPSPC          ;Skip over spaces
   28 003342  121327  000075                                CMPB    (R3),#'=        ;Value specified?
   29 003346  001005                                        BNE     3$              ;If not, allow unlimited
   30                                               ;
   31                                               ;   Accrue the parameter value
   32                                               ;
   33 003350  004767  000000G                       1$:     CALL    ACRDEC          ;Accrue decimal parameter
   34 003354  020127  000177                                CMP     R1,#127.        ;Asked for more than max?
   35 003360  101402                                        BLOS    2$              ;If not, set the requested number
   36 003362  012701  177777                        3$:     MOV     #-1,R1          ;Allow unlimited scrolling
   37                                               ;
   38                                               ;   Store value into EMT argument block
   39                                               ;
   40 003366  110167  174501                        2$:     MOVB    R1,WINMAK+5     ;Set max scroll lines
   41                                               ;
   42                                               ;   Finished
   43                                               ;
   44 003372  012601                                        MOV     (SP)+,R1
   45 003374  000207                                        RETURN
   46
   47                                               ;-------------------------------------------------------------------------
   48                                               ;   SET WINDOW/NOSCROLL
   49                                               ;
   50 003376  105067  174471                        WINNOS: CLRB    WINMAK+5        ;0 scroll lines ==> No scrolling
   51 003402  000207                                        RETURN
   52
   53                                               ;-------------------------------------------------------------------------
   54                                               ;   Process a SET WINDOW option that just sets a flag bit
   55                                               ;
   56 003404  051467  174406                        WINBIT: BIS     (R4),WINFLG     ;Set appropriate flag bit
   57 003410  000207                                        RETURN
```

```
    1                                          . SBTTL  .         Printwindow
    2                                 ; -------------------------------------------------------------------
    3                                 ;   SET PRINTWINDOW/DEVICE=device/TYPE=type
    4                                 ;
    5 003412                         SETPRS:
    6                                 ;
    7                                 ;   Process qualifiers specified with command
    8                                 ;
    9 003412  012704  001574'                 MOV     #PRSHD, R4      ;Point to option driver list
   10 003416  004767  000000G                 CALL    SCNOPS          ;Process all of the command options
   11                                 ;
   12                                 ;   Finished
   13                                 ;
   14 003422  000167  000000G                 JMP     RDCMD
   15                                 ;
   16                                 ;   Process the /DEVICE=device qualifer
   17                                 ;
   18 003426  010246                 PRSDEV: MOV     R2, -(SP)
   19 003430  052767  000000G 000000G         BIS     #PA$FLG, JPWFLG ;Default to using flag pages on new device
   20 003436  004767  000000G                 CALL    CHKEQ           ;Equal sign should follow qualifier name
   21 003442  004767  000000G                 CALL    GTRD50          ;Accrue the device name
   22 003446  122327  000072                  CMPB    (R3)+, #':      ;Colon specified with device name?
   23 003452  001401                          BEQ     1$              ;Br if yes
   24 003454  005303                          DEC     R3              ;Backup pointer
   25 003456  016767  000000G 000000G 1$:     MOV     R50BUF, JPWDEV  ;Store into job context cell
   26 003464  012602                          MOV     (SP)+, R2
   27 003466  000207                          RETURN
   28                                 ;
   29                                 ;   Process the /TYPE=type qualifier
   30                                 ;
   31 003470  010446                 PRSTYP: MOV     R4, -(SP)
   32 003472  004767  000000G                 CALL    CHKEQ           ;Make sure we have an equal sign
   33 003476  012704  001670'                 MOV     #PWTHD, R4      ;Point to table of printer types
   34 003502  004767  000000G                 CALL    SEARCH          ;Accrue and look up printer name
   35 003506  103406                          BCS     1$              ;Br if invalid printer type
   36 003510  111467  000000G                 MOVB    (R4), JPWTYP    ;Set printer type code
   37 003514  016467  000002 000000G          MOV     2(R4), JPWFLG   ;Set printer attribute flags
   38 003522  000417                          BR      9$
   39 003524  005704                 1$:     TST     R4              ;Printer unrecognized or ambiguous?
   40 003526  001407                          BEQ     2$              ;Br if not recognized
   41 003530                                  FERR    #EM$PTA         ;Printer type ambigious
   42 003544  000406                          BR      9$
   43 003546                         2$:     FERR    #EM$PTU         ;Printer type unrecognized
   44                                 ;
   45                                 ;   Finished
   46                                 ;
   47 003562  012604                 9$:     MOV     (SP)+, R4
   48 003564  000207                          RETURN
   49                                 ;
   50                                 ;   Process the /LETTERQUALITY qualifier
   51                                 ;
   52 003566  052767  000000G 000000G PRSLET: BIS     #PA$LET, JPWFLG ;Set letter-quality flag
   53 003574  000207                          RETURN
   54                                 ;
   55                                 ;   Process the /DRAFT qualifier
   56                                 ;
   57 003576  042767  000000G 000000G PRSDRF: BIC     #PA$LET, JPWFLG ;Clear letter-quality flag
```

```
 58 003604  000207                             RETURN
 59                                        ;
 60                                        ;  Process the /BELL qualifier
 61                                        ;
 62 003606  052767  000000G 000000G PRSBEL: BIS     #PA$BEL,JPWFLG  ;Set bell flag
 63 003614  000207                             RETURN
 64                                        ;
 65                                        ;  Process the /NOBELL qualifier
 66                                        ;
 67 003616  042767  000000G 000000G PRSNBL: BIC     #PA$BEL,JPWFLG  ;Clear bell flag
 68 003624  000207                             RETURN
 69                                        ;
 70                                        ;  Process the /FLAG qualifier
 71                                        ;
 72 003626  052767  000000G 000000G PRSFLG: BIS     #PA$FLG,JPWFLG  ;Set flagpage flag
 73 003634  000207                             RETURN
 74                                        ;
 75                                        ;  Process the /NOFLAG qualifier
 76                                        ;
 77 003636  042767  000000G 000000G PRSNFL: BIC     #PA$FLG,JPWFLG  ;Clear flagpage flag
 78 003644  000207                             RETURN
 79                                        ;
 80                                        ;  Process the /STAMP qualifier
 81                                        ;
 82 003646  052767  000000G 000000G PRSSTM: BIS     #PA$DTS,JPWFLG  ;Set date/time stamp flag
 83 003654  000207                             RETURN
 84                                        ;
 85                                        ;  Process the /NOSTAMP qualifier
 86                                        ;
 87 003656  042767  000000G 000000G PRSNST: BIC     #PA$DTS,JPWFLG  ;Clear date/time stamp flag
 88 003664  000207                             RETURN
 89                                        ;
 90                                        ;  Process the /WIDTH qualifier
 91                                        ;
 92 003666  042767  000000G 000000G PRSWID: BIC     #PA$NWD,JPWFLG  ;Clear nowidth flag
 93 003674  000207                             RETURN
 94                                        ;
 95                                        ;  Process the /NOWIDTH qualifier
 96                                        ;
 97 003676  052767  000000G 000000G PRSNWD: BIS     #PA$NWD,JPWFLG  ;Set nowidth flag
 98 003704  000207                             RETURN
 99                                        ;
100                                        ;  Process the /KEYPRINT qualifier
101                                        ;
102 003706  116701  000000G         PRSKEY: MOVB    CORUSR,R1        ;Get job index number
103 003712  052761  000000G 000000G         BIS     #$PWKEY,LSW11(R1) ;Set KEYPRINT flag
104 003720  000207                             RETURN
105                                        ;
106                                        ;  Process the /NOKEYPRINT qualifier
107                                        ;
108 003722  116701  000000G         PRSNKY: MOVB    CORUSR,R1        ;Get job index number
109 003726  042761  000000G 000000G         BIC     #$PWKEY,LSW11(R1) ;Clear KEYPRINT flag
110 003734  000207                             RETURN
```

```
    1                                          .SBTTL .          Priority
    2                                  ;-------------------------------------------------------------------
    3                                  ;   SET PRIORITY value
    4                                  ;
    5 003736  004767  000000G          SETPRI: CALL     ACRDEC           ;Accrue decimal value
    6 003742  020127  000000G                  CMP      R1,#MAXPRI       ;Is priority too large?
    7 003746  101425                           BLOS     3$               ;Br if ok
    8 003750                           5$:      FERR     #BADPRI          ;Invalid priority value
    9 003764  012705  000000                   MOV      #0,R5            ;Print minimum priority
   10 003770  004767  000000G                  CALL     PRTDEC
   11 003774                                   .PRINT   #TOTXT           ;Print " to "
   12 004002  116705  000000G                  MOVB     MXJPRI,R5        ;Print maximum priority
   13 004006  004767  000000G                  CALL     PRTDEC
   14 004012                                   .PRINT   #CRLF            ;End print line
   15 004020  000427                           BR       9$
   16 004022  120167  000000G          3$:      CMPB     R1,MXJPRI        ;Does it exceed max allowed for job?
   17 004026  101417                           BLOS     2$               ;Br if ok
   18 004030                                   FERR     #HIPRI           ;Priority is too high
   19 004044  116705  000000G                  MOVB     MXJPRI,R5        ;Get max allowed value
   20 004050  004767  000000G                  CALL     PRTDEC           ;Display max priority
   21 004054                                   .PRINT   #CRLF
   22 004062  116701  000000G                  MOVB     MXJPRI,R1        ;Get max allowed for job
   23 004066  012700  000062'          2$:      MOV      #PRIEMT,R0       ;Point to EMT arg block to set job priority
   24 004072  010160  000002                   MOV      R1,2(R0)         ;Store prio value into EMT arg block
   25 004076  104375                           EMT      375              ;Set job priority
   26 004100  000167  000000G          9$:      JMP      RDCMD            ;Go get next command
```

```
    1                                                .SBTTL .        Maxpriority
    2                                          ; --------------------------------------------------------------------------
    3                                          ;   SET MAXPRIORITY value
    4                                          ;
    5 004104  004767  000000G        SETMPR:  CALL    ACRDEC          ;Accrue decimal value
    6 004110  020127  000000G                 CMP     R1,#MAXPRI      ;Larger than max allowed?
    7 004114  101425                           BLOS    1$             ;Br if ok
    8 004116                                   FERR    #BADPRI         ;Invalid priority value
    9 004132  005000                           CLR     R0             ;Get minimum value
   10 004134  004767  000000G                 CALL    PRTDEC          ;Print minimum value
   11 004140                                   .PRINT  #TOTXT          ;Print " to "
   12 004146  012705  000000G                 MOV     #MAXPRI,R5      ;Get maximum value
   13 004152  004767  000000G                 CALL    PRTDEC          ;Print it
   14 004156                                   .PRINT  #CRLF
   15 004164  116701  000000G                 MOVB    VPRIDF,R1       ;Set to default value
   16 004170  120167  000000G        1$:      CMPB    R1,MXJPRI       ;Is he attempting to increase the max pri?
   17 004174  101404                           BLOS    2$             ;Br if not
   18 004176                                   FABORT  #EM$CIP         ;Cannot increase maxpri
   19 004206  110167  000000G        2$:      MOVB    R1,MXJPRI       ;Set maximum job priority
   20 004212  116702  000000G                 MOVB    CORUSR,R2       ;Get current job index number
   21 004216  126201  000000G                 CMPB    LBSPRI(R2),R1   ;Is current priority too high?
   22 004222  101405                           BLOS    3$             ;Br if ok
   23 004224  012700  000062'                 MOV     #PRIEMT,R0      ;Point to EMT arg block to set prio
   24 004230  010160  000002                  MOV     R1,2(R0)        ;Set new priority for job
   25 004234  104375                           EMT     375
   26 004236  000167  000000G        3$:      JMP     RDCMD           ;Finished
```

```
 1                                          .SBTTL   .        Error
 2                                      ;---------------------------------------------------------------------
 3                                      ;   ROUTINE TO SET ERROR SEVERITY ABORT LEVEL
 4                                      ;
 5 004242  111467  000000G             SETSEV: MOVB    @R4,ERRSEV         ;SET ERROR SEVERITY FOR USER
 6 004246  111404                              MOVB    @R4,R4             ;GET ERROR SEVERITY LEVEL
 7 004250  042704  177400                      BIC     #177400,R4         ;CLEAR HIGH ORDER BYTE
 8 004254  052704  001400                      BIS     #3*400,R4          ;SET COMMAND FILE NESTING DEPTH
 9 004260                                      .PVAL   #XAREA,#<CFABLV-MONVEC>,R4 ;SET ABORT LEVEL IN SIMUL RMON
10 004304  000167  000000G                     JMP     RDCMD
11
12                                          .SBTTL   .        Shutdown
13                                      ;---------------------------------------------------------------------
14                                      ;  Set the flag that says a system shutdown is being done.
15                                      ;
16 004310  004767  000000G             SETSHT: CALL    CKPRIV             ;User must have privilege
17 004314  112767  177777  000000G             MOVB    #-1,STPFLG         ;Say system shutdown taking place
18 004322  005067  000000G                     CLR     BOTDEV             ;Reboot from system disk
19 004326  000167  000000G                     JMP     RDCMD
20
21                                      ;---------------------------------------------------------------------
22                                      ;  Clear the flag that says a shutdown is taking place.
23                                      ;
24 004332  004767  000000G             SETNSH: CALL    CKPRIV             ;User must have privilege
25 004336  105067  000000G                     CLRB    STPFLG             ;Say we are not doing a shutdown
26 004342  000167  000000G                     JMP     RDCMD
```

```
    1                                              .SBTTL  .        Log
    2                                      ;-------------------------------------------------------------------
    3                                      ; Process the SET LOG command.
    4                                      ;
    5 004346                              SETLOG:
    6                                      ;
    7                                      ; Accrue option keyword
    8                                      ;
    9 004346  012704  002350'                     MOV     #STLGHD,R4        ;Point to keyword table
   10 004352  004767  000000G                     CALL    SCNOPS           ;Process command qualifiers
   11 004356  000167  000000G                     JMP     RDCMD            ;Finished with command
   12                                      ;
   13                                      ; SET LOG CLOSE
   14                                      ;
   15 004362  004767  000000G             STLGCL: CALL    LOGCLS           ;Close the log file
   16 004366  000207                              RETURN
   17                                      ;
   18                                      ; SET LOG CLEAN
   19                                      ;
   20 004370  010346                      STLGCN: MOV     R3,-(SP)
   21 004372  032767  000000G 000000G             BIT     #LF$OPN,LOGFLG   ;Is the log file open now?
   22 004400  001427                              BEQ     9$               ;Br if not
   23 004402  012767  000000G 000000G             MOV     #LOGBUF,LOGPTR   ;Reset buffer pointer
   24 004410  005067  000000G                     CLR     LOGBLK           ;Start writing at block 0
   25 004414  012703  000000G                     MOV     #LOGCHN,R3       ;Get channel # used for log file
   26 004420  016700  000000G                     MOV     CXTRMN,R0        ;Point to simulated RMON area
   27 004424  062700  000000G                     ADD     #R$CHN,R0        ;Point to channel block for channel 0
   28 004430  020327  000021                      CMP     R3,#17.          ;Is this channel in extended chan area?
   29 004434  103404                              BLO     1$               ;Br if not
   30 004436  162703  000021                      SUB     #17.,R3          ;Get channel # relative to extended channels
   31 004442  062700  000000C                     ADD     #R$XCHN-R$CHN,R0 ;Point to 1st extended channel block
   32 004446  070327  000000G             1$:     MUL     #CHNSIZ,R3       ;Get offset to block of interest
   33 004452  060003                              ADD     R0,R3            ;Point to our channel block
   34 004454  005063  000000G                     CLR     C.USED(R3)       ;Say no data has been written to file
   35 004460  012603                      9$:     MOV     (SP)+,R3
   36 004462  000207                              RETURN
   37                                      ;
   38                                      ; SET LOG WRITE
   39                                      ;
   40 004464  052767  000000G 000000G     STLGWR: BIS     #LF$WRT,LOGFLG   ;Enable writes to log file
   41 004472  000207                              RETURN
   42                                      ;
   43                                      ; SET LOG NOWRITE
   44                                      ;
   45 004474  042767  000000G 000000G     STLGNW: BIC     #LF$WRT,LOGFLG   ;Disable log file
   46 004502  000207                              RETURN
   47                                      ;
   48                                      ; SET LOG INPUT
   49                                      ;
   50 004504  042767  000000G 000000G     STLGIN: BIC     #LF$OUT,LOGFLG   ;Clear output-logging flag
   51 004512  052767  000000G 000000G             BIS     #LF$IN,LOGFLG    ;Set input-logging flag
   52 004520  000207                              RETURN
   53                                      ;
   54                                      ; SET LOG OUTPUT
   55                                      ;
   56 004522  042767  000000G 000000G     STLGOT: BIC     #LF$IN,LOGFLG    ;Clear input-logging flag
   57 004530  052767  000000G 000000G             BIS     #LF$OUT,LOGFLG   ;Set output-logging flag
```

```
    58 004536  000207                                    RETURN
    59                                           ;
    60                                           ;   SET LOG ALL
    61                                           ;
    62 004540  052767   0000000 0000000 STLGAL:  BIS     #LF$IN!LF$OUT,LOGFLG ;Log both input and output
    63 004546  000207                                    RETURN
```

```
     1                                          ;
     2                                          ;   SET LOG FILE=file-spec
     3                                          ;
     4 004550   010446                  STLGFL: MOV     R4,-(SP)
     5 004552   010546                          MOV     R5,-(SP)
     6 004554   004767  000000G                 CALL    LOGCLS          ;Close current log file
     7 004560   004767  000000G                 CALL    SKPSPC          ;Skip over spaces
     8 004564   122327  000075                  CMPB    (R3)+,#'=        ;See if equal sign was specified
     9 004570   001401                          BEQ     1$              ;Br if yes (skip over it)
    10 004572   005303                          DEC     R3              ;Point back to first char of file spec
    11                                          ;
    12                                          ;   Accrue the file spec
    13                                          ;
    14 004574   012704  000000G         1$:     MOV     #R50LOG,R4      ;Point to default extension ("LOG")
    15 004600   012705  000001                  MOV     #1,R5           ;Tell ACRFIL this is an output file
    16 004604   004767  000000G                 CALL    ACRFIL          ;Accrue the file spec
    17 004610   103514                          BCS     10$             ;Br if invalid file spec
    18                                          ;
    19                                          ;   Translate logical device name to physical
    20                                          ;
    21 004612   012705  000000G                 MOV     #FILNAM,R5      ;Point to file name buffer
    22 004616   004767  000000G                 CALL    LOGASN          ;Perform logical name translation
    23                                          ;
    24                                          ;   Check for empty file name
    25                                          ;
    26 004622   012767  000000G 000222          MOV     #DS$DIR,DSBLOK+0 ;Assume device is file struct
    27                                 ;This forces it to check for file name if device is unrecognized by .DSTAT
    28 004630                                   .DSTAT  #DSBLOK,#FILNAM ;Find out what kind of device it is
    29 004642   032767  000000G 000202          BIT     #DS$DIR,DSBLOK+0 ;Standard file structured device?
    30 004650   001403                          BEQ     5$              ;If not, can be opened without file name
    31 004652   005767  000002G                 TST     FILNAM+2        ;Was a file name specified?
    32 004656   001471                          BEQ     10$             ;Error if not file name on dir struct device
    33                                          ;
    34                                          ;   Don't allow log output to go to "TT:"
    35                                          ;
    36 004660   016700  000000G         5$:     MOV     FILNAM,R0       ;Get the log device name
    37 004664   004767  000000G                 CALL    CHKTTD          ;Is the log device TT?
    38 004670   103464                          BCS     10$             ;Br if yes -- Error
    39                                          ;
    40                                          ;   Try to open the file
    41                                          ;
    42 004672                                   .ENTER  #XAREA,#LOGCHN,#FILNAM,FILNAM+8.
    43 004720   103004                          BCC     2$              ;Br if enter ok
    44 004722                                   FABORT  #BDLGOP         ;Cannot open log file
    45                                          ;
    46                                          ;   Save information about the log file physical device
    47                                          ;
    48 004732   016705  000000G         2$:     MOV     FILNAM,R5       ;Get log file device name
    49 004736   004767  000000G                 CALL    CHKDEV          ;Convert to dev # and unit #
    50 004742   103421                          BCS     3$              ;Br if invalid device
    51 004744   020467  000000G                 CMP     R4,LDDEVX       ;Is log file on a logical disk?
    52 004750   001407                          BEQ     4$              ;Br if yes
    53 004752   110467  000000G                 MOVB    R4,LOGDVU       ;Save log file device #
    54 004756   110067  000001G                 MOVB    R0,LOGDVU+1     ;Save log file unit #
    55 004762   005067  000000G                 CLR     LOGBAS          ;Say not on a logical disk
    56 004766   000407                          BR      3$
    57 004770   006300                  4$:     ASL     R0              ;Convert unit # to word table index
```

```
   58 004772 016067  000000G 000000G           MOV      LDPDEV(RO),LOGDVU ;Save physical device and unit #
   59 005000 016067  000000G 000000G           MOV      LDBASE(RO),LOGBAS ;Save base block # of logical disk
   60                                       ;
   61                                       ;   Set "hold" mode for spooled log files
   62                                       ;
   63 005006 012700  000000G           3$:   MOV      #SPLHLA,RO        ;Point to emt argument block
   64 005012 104375                          EMT      375               ;Set hold mode for log file
   65                                       ;
   66                                       ;   Initialize the log file
   67                                       ;
   68 005014 052767  000000C 000000G           BIS      #<LF$OPN!LF$WRT!LF$IN!LF$OUT>,LOGFLG ;Say log file is open
   69 005022 012767  000000G 000000G           MOV      #LOGBUF,LOGPTR    ;Init pointer to log buffer
   70 005030 005067  000000G                  CLR      LOGBLK            ;Write to block 0
   71                                       ;
   72                                       ;   Finished
   73                                       ;
   74 005034 012605                          MOV      (SP)+,R5
   75 005036 012604                          MOV      (SP)+,R4
   76 005040 000207                          RETURN
   77                                       ;
   78                                       ;   Error -- Invalid file spec
   79                                       ;
   80 005042                          10$:   FABORT   #BDFNAM           ;Invalid file name
   81 005052 000000  000000  000000  DSBLOK: .WORD    0,0,0,0,0         ;.DSTAT result block
      005060 000000  000000
```

```
    1                                          .SBTTL .        Logoff
    2                               ;-----------------------------------------------------------------
    3                               ;  Process the SET LOGOFF command.
    4                               ;  This command is used to declare a logoff command file.
    5                               ;
    6 005064                       SETLOF:
    7                               ;
    8                               ;  Accrue option keyword
    9                               ;
   10 005064   012704   002414'            MOV     #STLOHD,R4      ;Point to keyword table
   11 005070   004767   000000G            CALL    SEARCH          ;Identify keyword
   12 005074   103002                      BCC     1$              ;Br if identified keyword
   13 005076   000167   173352             JMP     BDSO            ;Invalid option
   14                               ;
   15                               ;  Enter keyword processing routine
   16                               ;
   17 005102   000134            1$:       JMP     @(R4)+          ;Enter keyword processing routine
   18                               ;
   19                               ;  SET LOGOFF FILE=file-spec
   20                               ;
   21 005104   032761   000000G 000000G STLOFL: BIT  #$SUCF,LSW9(R1) ;Are we in a startup command file?
   22 005112   001010                      BNE     1$              ;Br if yes
   23 005114   032767   000000G 000000G      BIT   #PO$SYS,PRIVCO  ;Do we have SYSPRV privilege?
   24 005122   001004                      BNE     1$              ;Br if yes
   25 005124                               FABORT  #EM$NSF         ;Not in a startup command file
   26                               ;
   27                               ;  Accrue the file spec
   28                               ;
   29 005134   004767   000000G   1$:       CALL    SKPSPC          ;Skip over spaces
   30 005140   122327   000075             CMPB    (R3)+,#'=        ;Should have equal sign
   31 005144   001401                      BEQ     3$              ;Br if yes (skip over it)
   32 005146   005303                      DEC     R3              ;Point to 1st char of file name
   33 005150   012704   000000G   3$:       MOV     #R50COM,R4      ;Set "COM" as default extension
   34 005154   005005                      CLR     R5              ;Tell ACRFIL this is an input file
   35 005156   004767   000000G            CALL    ACRFIL          ;Accrue the file spec
   36 005162   103416                      BCS     10$             ;Br if invalid file spec
   37                               ;
   38                               ;  Translate logical file device name to physical device
   39                               ;
   40 005164   012705   000000G            MOV     #FILNAM,R5      ;Point to file spec
   41 005170   004767   000000G            CALL    LOGASN          ;Perform any logical device assignment
   42                               ;
   43                               ;  Move file spec to job context block
   44                               ;
   45 005174   012705   000000G            MOV     #FILNAM,R5      ;Point to file spec
   46 005200   012704   000000G            MOV     #LOFSPC,R4      ;Point to job context area
   47 005204   012700   000004             MOV     #4,R0           ;Get # words to move
   48 005210   012524            2$:       MOV     (R5)+,(R4)+     ;Move file spec to job context block
   49 005212   077002                      SOB     R0,2$
   50                               ;
   51                               ;  Finished
   52                               ;
   53 005214   000167   000000G            JMP     RDCMD
   54                               ;
   55                               ;  Invalid file spec
   56                               ;
   57 005220                       10$:      FABORT  #BDFNAM         ;Invalid file name
```

```
  1                                                  .SBTTL .          Subprocess
  2                                          ;-----------------------------------------------------------------
  3                                          ; Process the SET SUBPROCESS command.
  4                                          ; This command is used to declare a command file to be executed when
  5                                          ; a sub process (virtual line) is initiated.
  6                                          ;
  7 005230                                   SETSBP:
  8                                          ;
  9                                          ;  Accrue option keyword
 10                                          ;
 11 005230  012704  002424'                          MOV     #SBPHD,R4          ;Point to keyword table
 12 005234  004767  000000G                          CALL    SCNOPS             ;Process the options
 13 005240  000167  000000G                          JMP     RDCMD
 14                                          ;
 15                                          ;  SET SUBPROCESS/FILE=file-spec
 16                                          ;
 17 005244  010446                          SBPFIL: MOV     R4,-(SP)
 18 005246  010546                                  MOV     R5,-(SP)
 19                                          ;
 20                                          ;  This command is only legal within a start-up command file
 21                                          ;
 22 005250  032761  000000G 000000G                 BIT     #$SUCF,LSW9(R1)    ;Are we in a startup command file?
 23 005256  001010                                  BNE     1$                 ;Br if yes
 24 005260  032767  000000G 000000G                 BIT     #PO$SYS,PRIVCO     ;Do we have SYSPRV privilege?
 25 005266  001004                                  BNE     1$                 ;Br if yes
 26 005270                                          FABORT  #EM$NSF            ;Not in a startup command file
 27                                          ;
 28                                          ;  Accrue the file spec
 29                                          ;
 30 005300  004767  000000G                 1$:     CALL    ACRTXT             ;Accrue the file string
 31 005304  020027  000017                          CMP     RO,#15.            ;Compare with max legal length
 32 005310  101404                                  BLOS    3$                 ;Br if ok
 33 005312                                          FABORT  #EM$STL            ;File spec is too long
 34                                          ;
 35                                          ;  Move file spec to job context block
 36                                          ;
 37 005322  012705  000000G                 3$:     MOV     #BLKO,R5           ;Point to file spec
 38 005326  012704  000000G                         MOV     #SBPSUF,R4         ;Point to job context area
 39 005332  112524                          2$:     MOVB    (R5)+,(R4)+        ;Move file spec to job context block
 40 005334  001376                                  BNE     2$                 ;Loop till all moved
 41                                          ;
 42                                          ;  Finished
 43                                          ;
 44 005336  012605                                  MOV     (SP)+,R5
 45 005340  012604                                  MOV     (SP)+,R4
 46 005342  000207                                  RETURN
 47                                          ;
 48                                          ;  Invalid file spec
 49                                          ;
 50 005344                                  10$:    FABORT  #BDFNAM            ;Invalid file name
```

```
     1                                                    .SBTTL .         Kmon
     2                                          ;------------------------------------------------------------------
     3                                          ;  Process the SET KMON USER command.
     4                                          ;  Verify that a user-written command interface program exists and if so,
     5                                          ;  set a flag saying to use it for this job.
     6                                          ;
     7 005354                                   SETUKM:
     8                                          ;
     9                                          ;  See if user specified a file spec
    10                                          ;
    11 005354   004767   000000G                      CALL    SKPSPC              ;Skip over any spaces
    12 005360   105713                                TSTB    (R3)                ;Anything specified following "USER"?
    13 005362   001417                                BEQ     2$                  ;Br if not
    14 005364   122327   000075                        CMPB    (R3)+,#'=          ;Equal sign?
    15 005370   001053                                BNE     3$                  ;Error if not
    16                                          ;
    17                                          ;  Accrue the user file name
    18                                          ;
    19 005372   012704   000000G                      MOV     #R50SAV,R4          ;Set default extension
    20 005376   005005                                CLR     R5                  ;Tell ACRFIL this is an input file
    21 005400   004767   000000G                      CALL    ACRFIL              ;Accrue the file spec
    22 005404   103451                                BCS     4$                  ;Br if invalid file spec
    23 005406   012704   000000G                      MOV     #FILNAM,R4          ;Point to area with file spec
    24 005412   010405                                MOV     R4,R5
    25 005414   004767   000000G                      CALL    LOGASN              ;Convert logical device name to physical
    26 005420   000404                                BR      6$
    27                                          ;
    28                                          ;  User did not specify a file name.  Default to "SY:UKMON.SAV".
    29                                          ;
    30 005422   012704   000000G               2$:    MOV     #UCIDEF,R4          ;Point to default file spec
    31 005426   016714   000000G                      MOV     SYNAME,(R4)         ;Set physical SY device
    32                                          ;
    33                                          ;  Move file spec to UCISPC area in context block
    34                                          ;
    35 005432   012700   000004               6$:    MOV     #4,R0               ;Get # words to move
    36 005436   012705   000000G                      MOV     #UCISPC,R5          ;Store UCI file spec here
    37 005442   012425               5$:    MOV     (R4)+,(R5)+         ;Move file spec to job context block
    38 005444   077002                                SOB     R0,5$
    39                                          ;
    40                                          ;  Try to lookup the specified file
    41                                          ;
    42 005446                                        .LOOKUP #XAREA,#1,#UCISPC;Try to find command processor program
    43 005466   103410                                BCS     1$                  ;Br if not available
    44 005470                                         .CLOSE  #1                  ;Close the program file
    45 005476   052761   000000G 000000G             BIS     #$UKMON,LSW7(R1);Set flag saying to use user-written program
    46 005504   000167   000000G                      JMP     RDCMD
    47                                          ;
    48                                          ;  Error processing
    49                                          ;  Cannot find specified file
    50                                          ;
    51 005510                                  1$:    FABORT  #EM$NUK             ;User-written command processor not available
    52                                          ;
    53                                          ;  Invalid syntax
    54                                          ;
    55 005520                                  3$:    FABORT  #MISSEQ
    56                                          ;
    57                                          ;  Invalid file spec
```

```
   58                                      ;
   59 005530                               4$:      FABORT   #BDFNAM
```

```
    1                                    ;-------------------------------------------------------------------
    2                                    ; Process the SET KMON NEW command.
    3                                    ; Begin to use a new version of TSKMON.
    4                                    ;
    5 005540   004767   0000000  SETKNW: CALL    CKSYPV          ;Must have SYSPRV privilege
    6 005544                             .LOOKUP #XAREA,#1,#KMNNAM ;TRY TO LOOKUP SY:TSKMON.SAV
    7 005564   103500                    BCS     1$              ;BR IF NOT THERE
    8                                    ;
    9                                    ; Get information about TSKMON out of block 0 of file
   10                                    ;
   11 005566                             .READW  #XAREA,#1,#BLK0,#256.,#0 ;READ IN BLOCK 0
   12 005624   016700   0000500          MOV     BLK0+50,R0      ;GET TOP ADDRESS OF KMON
   13 005630   062700   000003           ADD     #3,R0           ;BOUND UP TO NEXT WORD
   14 005634   042700   000001           BIC     #1,R0           ;FORCE EVEN
   15 005640   010067   0000000          MOV     R0,KMNTOP
   16 005644   162700   0000000          SUB     #KMNBAS,R0      ;BASE ADDRESS OF KMON
   17 005650   010067   0000000          MOV     R0,KMNHI        ;TOP OF TSKMON-KMNBAS
   18 005654   062700   000777           ADD     #511.,R0        ;BOUND UP TO PAGE SIZE
   19 005660   000241                    CLC
   20 005662   006000                    ROR     R0              ;CVT TO # WORDS
   21 005664   000300                    SWAB    R0              ;CVT TO # PAGES
   22 005666   042700   177400           BIC     #^C377,R0
   23 005672   062700   0000000          ADD     #CXTPAG,R0      ;# PAGES NEEDED TO JOB CONTEXT AREA
   24 005676   010067   0000000          MOV     R0,KMNPGS       ;# 256 WORD PAGES FOR KMON & CONTEXT
   25 005702   016767   0000420 0000000  MOV     BLK0+42,KMNSTK  ;INITIAL STACK POINTER
   26 005710   016767   0000400 0000000  MOV     BLK0+40,KMNSTR  ;STARTING ADDRESS
   27                                    ;
   28                                    ; Now do .SAVESTATUS so we can quickly reopen to kmon
   29                                    ;
   30 005716                             .SAVEST #XAREA,#1,#KMNCHN ;SAVE STATUS
   31                                    ;
   32                                    ; Now purge and reopen channel 17 to the new file (for overlays).
   33                                    ;
   34 005736                             .PURGE  #17             ;Purge channel 17 (overlay channel)
   35 005744                             .REOPEN #XAREA,#17,#KMNCHN ;Open channel 17 to new file
   36                                    ;
   37                                    ; Exit to reload Kmon
   38                                    ;
   39 005764                             .EXIT                   ;EXIT AND REENTER NEW VERSION OF TSKMON
   40                                    ;
   41                                    ; Error: Cannot find new Kmon file
   42                                    ;
   43 005766                     1$:     FABORT  #NOKMON         ;ERROR -- CAN'T FIND TSKMON
   44
   45                                    ;-------------------------------------------------------------------
   46                                    ; Process the SET CCL NEW command.
   47                                    ; Begin to use a enw version of CCL
   48                                    ;
   49 005776   004767   0000000  SETCNW: CALL    CKSYPV          ;Must have SYSPRV privilege
   50 006002                             .LOOKUP #XAREA,#1,#CCLNAM ;TRY TO LOOKUP SY:CCL.SAV
   51 006022   103412                    BCS     1$              ;BR IF CAN'T FIND IT
   52 006024                             .SAVEST #XAREA,#1,#CCLSAV ;SAVE STATUS
   53 006044   000167   0000000          JMP     RDCMD           ;FINISHED
   54 006050                     1$:     FABORT  #NOCCL          ;CAN'T FIND CCL.SAV
```

```
    1                                          .SBTTL .        LD
    2                                  ;-----------------------------------------------------------------------
    3                                  ;  Process SET LDn command
    4                                  ;
    5 006060  005002          SETLD:  CLR     R2              ;ASSUME UNIT # = 0
    6 006062  020067  0000000         CMP     R0,R50LD        ;IS UNIT "LD"?
    7 006066  001404                  BEQ     1$              ;BR IF YES
    8 006070  010002                  MOV     R0,R2           ;GET DEVICE NAME
    9 006072  166702  0000000         SUB     R50LD0,R2       ;SUBTRACT "LD0" TO GET UNIT # IN R2
   10 006076  006302                  ASL     R2              ;CONVERT TO WORD TABLE INDEX
   11                                  ;
   12                                  ;  Get the option word
   13                                  ;
   14 006100  012704  001264'  1$:    MOV     #LDOPHD,R4      ;POINT TO OPTION TABLE
   15 006104  004767  0000000         CALL    SEARCH          ;ACCRUE AND CHECK OPTION WORD
   16 006110  103004                  BCC     2$              ;BR IF FOUND
   17 006112                          FABORT  #CSIMS1         ;INVALID OPTION WORD
   18                                  ;
   19                                  ;  Jump off to option processing routine
   20                                  ;
   21 006122  000134           2$:    JMP     @(R4)+          ;ENTER PROCESSING ROUTINE
   22                                  ;
   23                                  ;  SET LD CLEAN
   24                                  ;
   25 006124  004767  0000000  SLDCLN: CALL   LDCLEN          ;CLEAN UP LOGICAL DISKS
   26 006130  000167  0000000         JMP     RDCMD
   27                                  ;
   28                                  ;  SET LDn WRITE
   29                                  ;
   30 006134  042762  0000000 0000000 SLDWRT: BIC    #LD$RON,LDFLAG(R2) ;CLEAR READ-ONLY FLAG
   31 006142  000167  0000000         JMP     RDCMD           ;FINISHED
   32                                  ;
   33                                  ;  SET LDn NOWRITE
   34                                  ;
   35 006146  052762  0000000 0000000 SLDNWR: BIS    #LD$RON,LDFLAG(R2) ;SET READ-ONLY FLAG
   36 006154  000167  0000000         JMP     RDCMD           ;FINISHED
   37                                  ;
   38                                  ;  SET LDn FREE
   39                                  ;
   40 006160  004767  000052   SLDFRE: CALL   LDDMT           ;DISMOUNT THIS LD
   41 006164  004767  0000000         CALL    LDCLEN          ;NOW RESET ALL LOGICAL DISK ASSIGNMENTS
   42 006170  000167  0000000         JMP     RDCMD           ;FINISHED
```

```
     1                                      ;
     2                                      ;   SET LD EMPTY
     3                                      ;   (Dismount all logical disks)
     4                                      ;
     5 006174                              SLDEMP:
     6                                      ;
     7                                      ;   Begin loop to close log files and deassign any assignments for each
     8                                      ;   logical disk
     9                                      ;
    10 006174  005002                              CLR     R2              ;Init index to 1st LD
    11 006176  004767  000102      1$:             CALL    LOGDEA          ;Close any log files and make deassignments
    12 006202  062702  000002              ADD     #2,R2           ;Get index for next LD
    13 006206  020227  000016              CMP     R2,#14.         ;Have we done all?
    14 006212  101771                              BLOS    1$              ;Loop if not
    15                                      ;
    16                                      ;   Dismount the all LD's
    17                                      ;
    18 006214  012700  000110'             MOV     #DMTALD,R0      ;Emt arg. block to dismount all LD's
    19 006220  104375                              EMT     375             ;Dismount all LD's
    20 006222  103003                              BCC     9$              ;Branch if all LD's dismounted
    21 006224                              .PRINT  #EDMALD         ;"Unable to dismount all logical disks."
    22                                      ;
    23                                      ;   Finished
    24                                      ;
    25 006232  000167  000000G     9$:             JMP     RDCMD
```

```
     1                                          ;------------------------------------------------------------------
     2                                          ;  Subroutine to dismount a logical disk.
     3                                          ;
     4                                          ;  Inputs:
     5                                          ;    R2 = LD unit index
     6                                          ;
     7 006236  010246               LDDMT:  MOV     R2,-(SP)
     8 006240  004767  000040               CALL    LOGDEA              ;Close any log files and make deassignments
     9                                          ;
    10                                          ;  Tell system to stop doing directory caching for device
    11                                          ;
    12 006244  112767  000001  000000G         MOVB    #1,SERFLG          ;Do .SERR to avoid abort for illegal device
    13 006252  012700  000000C               MOV     #DMTARG,RO         ;Point to EMT arg block
    14 006256  104375                         EMT     375                ;Tell system to stop doing caching
    15 006260  105067  000000G               CLRB    SERFLG             ;Do .HERR
    16                                          ;
    17                                          ;  Clean out LD info tables
    18                                          ;
    19 006264  005062  000000G               CLR     LDPDEV(R2)         ;No physical device assignment
    20 006270  072227  000002               ASH     #2,R2              ;Get index into LDNAME table
    21 006274  005062  000000G               CLR     LDNAME(R2)         ;No file name
    22                                          ;
    23                                          ;  Finished
    24                                          ;
    25 006300  012602               4$:     MOV     (SP)+,R2
    26 006302  000207                        RETURN
```

```
    1                                         ;-------------------------------------------------------------------------------
    2                                         ;
    3                                         ;   Subroutine to check for open log files and deassign any assignments
    4                                         ;
    5                                         ;   Inputs:
    6                                         ;     R2 = LD unit index
    7                                         ;
    8 006304  010246              LOGDEA: MOV     R2,-(SP)
    9 006306  010546                      MOV     R5,-(SP)
   10                                         ;
   11                                         ;   See if this LD is in use
   12                                         ;
   13 006310  010200                      MOV     R2,R0           ;Get LD unit index
   14 006312  072027  000002              ASH     #2,R0           ;Cvt to index into name table
   15 006316  005760  000000G             TST     LDNAME(R0)      ;Is this LD in use?
   16 006322  001413                      BEQ     9$              ;Br if not
   17                                         ;
   18                                         ;   Build name of LD
   19                                         ;
   20 006324  010205                      MOV     R2,R5           ;Get LD unit index
   21 006326  006205                      ASR     R5              ;Convert to unit number
   22 006330  066705  000000G             ADD     R5OLDO,R5       ;Add "LDO" to form unit name
   23 006334  010567  000000G             MOV     R5,MNTDEV       ;Save name of device being dismounted
   24                                         ;
   25                                         ;   Close the log file if it is on the device being dismounted
   26                                         ;
   27 006340  004767  000000G             CALL    LOGCHK          ;Close log file if it is on this LD
   28                                         ;
   29                                         ;   Do DEASSIGN of anything assigned to this LD
   30                                         ;
   31 006344  010500                      MOV     R5,R0           ;Get LD name
   32 006346  004767  000000G             CALL    DEADEV          ;Do deassign
   33                                         ;
   34                                         ;   Finished
   35                                         ;
   36 006352  012605              9$:     MOV     (SP)+,R5
   37 006354  012602                      MOV     (SP)+,R2
   38 006356  000207                      RETURN
```

```
    1                                                    .SBTTL    .          SL
    2                                           ;--------------------------------------------------------------------
    3                                           ;   SET SL
    4                                           ;
    5                                           ;   Inputs:
    6                                           ;    R3 = Pointer to parameters following "SET SL".
    7                                           ;
    8 006360                                    SETSL:
    9                                           ;
   10                                           ;   Process each of the command parameters
   11                                           ;
   12 006360  012704  001314'                             MOV      #SLEHD,R4          ;Point to list of qualifiers
   13 006364  004767  000000G                             CALL     SCNOPS            ;Process the qualifiers
   14                                           ;
   15                                           ;   Finished
   16                                           ;
   17 006370  000167  000000G                             JMP      RDCMD
   18
   19
   20                                           ;--------------------------------------------------------------------
   21                                           ;   SET SL ON
   22                                           ;
   23 006374  105767  000000G                  SLOON:  TSTB     VSLEDT             ;Is SL available?
   24 006400  001007                                   BNE      1$                 ;Br if yes
   25 006402                                           FWARN    #EM$NSL            ;SL not genned into system
   26 006416  000425                                   BR       9$
   27 006420  032761  000000C 000000G 1$:      BIT      #<VT100!VT200!VT52>,LTRMTP(R1) ;VT100, VT200 or VT52
   28 006426  001007                                   BNE      2$                 ;Br if yes
   29 006430                                           FWARN    #EM$SLT            ;Invalid terminal type
   30 006444  000412                                   BR       9$
   31 006446  052761  000000G 000000G 2$:      BIS      #$SLON,LSW7(R1)   ;Enable SL for this line
   32 006454  032761  000000G 000000G          BIT      #$SLKED,LSW7(R1) ;Is Ked mode wanted?
   33 006462  001403                                   BEQ      9$                 ;Br if not
   34 006464                                           .PRINT   #SLKDON            ;Enable terminal alternate keypad mode
   35 006472  000207                  9$:      RETURN
   36                                           ;
   37                                           ;--------------------------------------------------------------------
   38                                           ;   SET SL OFF
   39                                           ;
   40 006474                          SLOOFF: .PRINT   #SLKDOF            ;Return numeric keypad to normal mode
   41 006502  042761  000000G 000000G          BIC      #$SLON,LSW7(R1)  ;Disable SL for this line
   42 006510  000207                                   RETURN
   43                                           ;
   44                                           ;--------------------------------------------------------------------
   45                                           ;   SET SL TTY
   46                                           ;
   47 006512  052761  000000G 000000G SLOTTY: BIS       #$SLTTY,LSW7(R1);Enable SL for .TTYIN
   48 006520  000207                                   RETURN
   49                                           ;
   50                                           ;--------------------------------------------------------------------
   51                                           ;   SET SL NOTTY
   52                                           ;
   53 006522  042761  000000G 000000G SLONTT: BIC       #$SLTTY,LSW7(R1);Disable SL for .TTYIN
   54 006530  000207                                   RETURN
   55                                           ;
   56                                           ;--------------------------------------------------------------------
   57                                           ;   SET SL WIDTH=n
```

```
  58                                        ;
  59 006532                                 SLOWID: FWARN   #EM$SLW          ;Print warning message
  60 006546   012705   000000G                      MOV     #SLMXLN,R5       ;Get SL width
  61 006552   004767   000000G                      CALL    PRTDEC           ;Print the value
  62 006556                                         .PRINT  #CRLF            ;Terminate the line
  63 006564   000207                                RETURN
  64                                        ;
  65                                        ;------------------------------------------------------------------
  66                                        ;    SET SL LET
  67                                        ;
  68 006566   052761   000000G 000000G SLOLET: BIS     #$SLLET,LSW7(R1);Enable LET feature
  69 006574   000207                                RETURN
  70                                        ;
  71                                        ;------------------------------------------------------------------
  72                                        ;    SET SL NOLET
  73                                        ;
  74 006576   042761   000000G 000000G SLONLT: BIC     #$SLLET,LSW7(R1);Disable LET feature
  75 006604   000207                                RETURN
  76                                        ;
  77                                        ;------------------------------------------------------------------
  78                                        ;    SET SL KED
  79                                        ;
  80 006606   052761   000000G 000000G SLOKED: BIS     #$SLKED,LSW7(R1);Remember SL is in KED mode
  81 006614                                         .PRINT  #SLKDON          ;Enable terminal alternate keypad mode
  82 006622   000207                                RETURN
  83                                        ;
  84                                        ;------------------------------------------------------------------
  85                                        ;    SET SL RT11
  86                                        ;
  87 006624   042761   000000G 000000G SLORT:  BIC     #$SLKED,LSW7(R1);Remember SL is not in KED mode
  88 006632                                         .PRINT  #SLKDOF          ;Return numeric keypad to normal mode
  89 006640   000207                                RETURN
  90                                        ;
  91                                        ;------------------------------------------------------------------
  92                                        ;    Unimplemented SL option
  93                                        ;
  94 006642                                 SLOUNI: FWARN   #EM$UIO          ;Unimplemented option
  95 006656   000207                                RETURN
  96                                        ;
  97                                        ;------------------------------------------------------------------
  98                                        ;    Ignored option
  99                                        ;
 100 006660   000207                        SLONOP: RETURN
 101                                        ;
 102                                        ;------------------------------------------------------------------
 103                                        ;    SET RECALL NORMAL/REVERSE
 104                                        ;
 105 006662   012704   001444'             SETREC: MOV     #RCLHD,R4        ;Point to option table
 106 006666   004767   000000G                      CALL    SEARCH           ;Accrue and check options
 107 006672   103006                                BCC     1$               ;Br if valid option
 108 006674                                         FWARN   #INVOPT          ;Invalid option
 109 006710   111467   000000G             1$:      MOVB    @R4,RCLREV       ;Set into context cell
 110 006714   000167   000000G                      JMP     RDCMD
 111
 112          000001                                .END
Errors detected:  0
```

*** Assembler statistics

Work file reads: 0
Work file writes: 0
Size of work file: 12080 Words (48 Pages)
Size of core pool: 18176 Words (71 Pages)
Operating system: RT-11

Elapsed time: 00:01:56.07
,LP:TSKST1=DK:TSKST1/C/N:SYM

```
$1STLG     1-73
$8BIT      1-111
$AUTO      1-87
$CARUP     1-85
$CCLRN     1-86
$CFABT     1-106
$CFALL     1-112
$CFCCL     1-112
$CFDCC     1-112
$CFOPN     1-118
$CFSOT     1-110
$CHACT     1-61
$CLTST     1-96       5-45        5-46
$CTRLC     1-104
$CTRLD     1-154      10-38       10-40
$CTRLO     1-61
$CTRLS     1-91
$DBKMN     1-84       5-38        5-39
$DEAD      1-158
$DEBUG     1-155
$DEFER     1-124
$DETCH     1-89
$DIBOL     1-73       5-167       5-168
$DILUP     1-108
$DISCN     1-90
$DOOFF     1-114
$DUPRN     1-109
$ECHO      1-111
$EMTTR     1-95       5-174       5-175       10-48
$FORM      1-110
$FORMO     1-112
$HARD      1-158
$HITTY     1-72
$INCOR     1-128
$INDAB     1-159      5-62        5-63
$INDDF     1-157      5-32        5-33        14-86       14-96
$INDRN     1-157
$INIT      1-158
$INKMN     1-104
$KED       1-128      14-43       14-48       14-56
$KINIT     1-68
$LC        1-111
$LOFCF     1-203
$MLOCK     1-77
$NOIN      1-72       17-6
$NOINT     1-204
$NOWTT     1-72
$PAGE      1-111
$PHONE     1-158
$PRGLK     1-87
$PWKEY     1-32       20-103      20-109
$QTSET     1-133
$QUIET     1-125
$RNIOP     1-205
$SCOPE     1-111
$SGALL     1-124      5-206
```

```
$SGHIO   1-123    5-188    5-189
$SGIIO   1-123    5-190    5-191
$SGQO    1-105    5-192    5-193
$SGQ1    1-123    5-194    5-195
$SGQ1A   1-123    5-196    5-197
$SGQ1B   1-123    5-198    5-199
$SGQ1C   1-123    5-200    5-201
$SGQ2    1-123    5-202    5-203
$SGQ3    1-105    5-204    5-205
$SLKED   1-82    34-32    34-80    34-87
$SLLET   1-202   34-68    34-74
$SLON    1-202   34-31    34-41
$SLTTY   1-202   34-47    34-53
$SNWTT   1-155
$SPLJB   1-91
$STSNG   1-61
$SUCF    1-113   17-7     26-21    27-22
$SYSPS   1-160
$TAB     1-110
$TAPE    1-153
$TECO    1-133   14-43    14-49    14-55
$TTGAG   1-118
$UCLCF   1-119    5-53    14-61    14-68    14-74    14-79
$UCLCL   1-121    5-55    14-62    14-68    14-73    14-79
$UCLCM   1-121    5-54    14-62    14-67    14-74    14-79
$UCLRN   1-120
$UKMON   1-85     5-34    28-45
$VNOTT   1-48
$VTESC   1-81
$WILD    1-134    5-25     5-26
...V1    8-57     9-28     9-30    10-16    10-18    11-28    11-28    11-28    11-30    11-30    11-35    11-41
        11-41    11-44    12-74    12-75    12-75    12-95    12-96    12-96    12-97    12-97    13-5     13-5
        13-6     13-6     13-38    14-87    14-90    14-97    14-100   18-60    18-66    18-72    18-78    21-11
        21-14    21-21    22-11    22-14    23-9     25-28    25-42    25-42    25-42    28-42    28-42    28-42
        28-44    28-44    29-6     29-6     29-6     29-11    29-11    29-30    29-30    29-34    29-35    29-35
        29-50    29-50    29-50    29-52    29-52    31-21    34-34    34-40    34-62    34-81    34-88
...V2    8-57     8-57#   11-28    11-28    11-28#   11-28#   11-30    11-30    11-30#   11-30#   11-35    11-35#
        11-41    11-41    11-41#   11-41#   11-44    11-44#   12-74    12-74#   12-75    12-75#   12-75#
        12-95    12-95#   12-96    12-96    12-96#   12-96#   12-97    12-97    12-97#   12-97#   13-5     13-5
        13-5#    13-5#    13-6     13-6#    13-38    13-38#   14-90    14-90#   14-90#   14-100   14-100#  14-100#
        23-9     23-9#    23-9#    25-42    25-42    25-42#   25-42#   28-42    28-42    28-42#   28-42#   28-44
        28-44#   29-6     29-6     29-6#    29-6#    29-11    29-11    29-11#   29-11#   29-30    29-30    29-30#
        29-30#   29-34    29-34#   29-35    29-35    29-35#   29-35#   29-50    29-50    29-50#   29-50#   29-52
        29-52    29-52#   29-52#
...V3   14-90    14-90    14-90#   14-100   14-100   14-100#  23-9     23-9     23-9#
ABRTAD   1-48
ABRTCD   1-48
ABRTCF   1-165
ACRDEC   1-173    9-16     9-48    10-8     12-58    14-33    19-8     19-33    21-5     22-5
ACRFIL   1-172   25-16    26-35    28-21
ACRFN    1-165
ACROCT   1-175   12-60
ACRSPD   1-76
ACRSTR   1-37    15-22
ACRTXT   1-39    16-13    27-30
AD$$SZ   1-64
```

AD$DVU	1-64										
AD$JOB	1-64										
ADM3A	1-148										
ADM3FL	1-149										
ADM3NO	1-151										
AF$DBG	1-204										
AF$HIE	1-204										
AF$IOP	1-205										
AF$MEM	1-47										
AF$NOI	1-204										
AF$NOW	1-47										
AF$PLK	1-204										
AF$SCA	1-47										
ALCDEV	1-28										
ALCEND	1-63										
ALCHD1	1-60										
ALCHD2	1-60										
ALCTBL	1-63										
ALDBLK	1-188										
ALDEX	1-187	1-188									
ALFN	1-199										
AMBOPT	1-173	1-186	8-56								
AR$$SZ	1-202										
AR$CNT	1-201										
AR$CON	1-201										
AR$CPH	1-201										
AR$CPL	1-201										
AR$DMY	1-202										
AR$PRG	1-201										
AR$PRJ	1-201										
AR$UNM	1-201										
ARNRPB	1-202										
ASDEX	1-170										
ASKLNM	1-169										
ASNHD1	1-182										
ASNHD2	1-182										
ASNOVF	1-170	1-195									
ASNSRC	1-137	7-45									
AT$DEV	1-56	7-47									
AUTHFN	1-192										
BADCMD	1-169										
BADPMT	1-176	15-38									
BADPRI	1-176	21-8	22-8								
BADSAV	1-169										
BASMAP	1-142										
BDFNAM	1-172	25-80	26-57	27-50	28-59						
BDLGOP	1-177	25-44									
BDLIN	1-190										
BDSO.	8-44	8-54#	26-13								
BELL	2-7#										
BLANK	2-6#										
BLKO	1-168	11-30	12-23	12-39	13-5	13-28*	13-29*	13-30	15-25	16-23	27-37	29-11
	29-12	29-25	29-26								
BLKWDS	2-11#										
BOTDEV	1-103	23-18*									
BOTUNI	1-103										

```
C.CSW     1-115
C.DEVQ    1-115
C.SBLK    1-115
C.USED    1-41      24-34*
CALUCL    1-181
CASCBR    1-97
CASCUP    1-97
CASTBR    1-97
CASTBW    1-97
CASTRO    1-98
CASTWO    1-98
CCLHD     4-10      5-44#
CCLNAM    1-196     29-50
CCLSAV    1-94      29-52
CD$$SZ    1-116
CD$$UB    1-116
CD$BAS    1-116
CD$DVU    1-116
CD$JOB    1-116
CD$NAM    1-116
CD$TOP    1-133
CDBUF     1-38
CDGET     1-38
CF$IND    1-159     14-89     14-97
CF$QUT    1-159
CFABLV    1-160     23-9
CFBLK     1-125
CFBUF     1-94
CFCHAN    1-124
CFEND     1-94
CFHOLD    1-132
CFIND     1-114
CFLFL4    1-119
CFNEST    1-117
CFPNT     1-125
CFSEND    1-118
CFSP      1-118
CFSPND    1-131
CFSTK     1-68
CFSTS     1-159     14-87     14-90     14-97     14-100
CHAIN     1-104
CHKALC    1-111     7-75
CHKCLU    1-36      7-65
CHKDEV    1-196     25-49
CHKDLM    1-186
CHKEQ     1-45      20-20     20-32
CHKMNT    1-171
CHKMTX    1-171
CHKTTD    1-194     25-37
CHNSIZ    1-41      24-32
CINDAT    1-141
CINFLG    1-48
CKACOJ    1-45
CKCLUS    1-37
CKPRIV    1-173     14-6      14-14     23-16     23-24
CKSYPV    1-45      8-14      9-12      9-44      10-4      16-9      29-5      29-49
```

```
CKTERM    1-174
CL$COL    1-66
CL$EPN    1-41
CL$EPS    1-41
CL$LEN    1-101
CL$LIN    1-101
CL$LIX    1-62
CL$OPT    1-100
CL$RQH    1-63
CL$SKP    1-101
CL$WID    1-101
CL$WQH    1-63
CL$XLN    1-37
CLDEVX    1-139
CLEOFS    1-41
CLFREE    1-57
CLLINE    1-57
CLOTIR    1-65
CLRBIT    5-26      5-34      5-39      5-46      5-63      5-168     5-175     5-189     5-191     5-193     5-195     5-197
          5-199     5-201     5-203     5-205     5-206     14-15#
CLRPRV    1-43
CLSFEP    1-42
CLSFSP    1-98
CLTOTL    1-98      7-63
CLUNIT    1-57
CMDBUF    1-164     1-181
CMDCCL    1-196
CMDDSN    1-27
CMDFRM    1-27
CMDHD     1-26
CMDMEM    1-28
CMDOFF    1-26
CMDSET    1-28      7-5#
CMDSHO    1-28
CMDUSE    1-28
CMDWHO    1-28
CO$8BT    1-115
CO$BNI    1-100
CO$BNO    1-100
CO$CR     1-99
CO$CTL    1-99
CO$DEF    1-66
CO$DTR    1-98
CO$FF     1-99
CO$FFO    1-99
CO$LC     1-99
CO$LFI    1-100
CO$LFO    1-100
CO$TAB    1-99
COAD      1-187     1-188
COAF      1-200
COAL      1-187     1-188
COLOO     1-181
COMMA     2-10#
CONFG2    1-62
CONFIG    1-113
```

```
CORUSR    1-61      14-34     20-102    20-108    22-20
CPUAH     1-181
CPUAL     1-181
CR        2-5#
CRLF      1-176     9-30      10-18     21-14     21-21     22-14     34-62
CS$RON    1-80
CSHALC    1-103     10-9
CSHDEV    1-108
CSHDVN    1-108
CSHHD     1-163
CSHMSG    1-203
CSHSIZ    1-143
CSIMS1    1-175     13-39     30-17
CSIMS2    1-173     7-44      11-36
CSIMS4    1-195
CTDHD     4-12      5-236#
CTRLTT    1-102
CURMTX    1-186
CURPRM    1-130
CVDVNM    1-183
CVTTAB    1-164     7-5
CVTUC     1-163
CW$50H    1-113
CW$PRU    1-62
CXTBAS    1-107
CXTPAG    1-79      29-23
CXTRMN    1-41      24-26
CXTWDS    1-107
DATTIM    1-27
DCCRD     1-137
DCCWR     1-137
DCHNEW    6-33#     10-26
DCTRD     1-137
DCTWR     1-137
DEADEV    1-171     33-32
DELSPC    1-185
DETARG    1-200
DETHD     1-200
DETTXT    1-179
DEVHD1    1-182
DEVIDL    1-187     1-187     1-188
DEVUNT    1-174     11-20*    12-85
DFJMEM    1-68
DIABFL    1-125
DIABLO    1-147
DIABNO    1-126
DIVIDE    1-180
DIVSOR    1-193
DJABMS    1-191
DKASHD    1-59
DKSAV     1-164
DLCEMT    1-28
DLMSG     1-191
DLTXT     1-178
DMTALD    6-60#     31-18
DMTALL    1-191
```

```
DMTARG    1-170     32-13
DMTSUB    1-196
DOASGN    1-85
DORUN     1-27
DOSTOP    1-192
DS$DIR    1-204     25-26     25-29
DSBLOK    25-26*    25-28     25-29     25-81#
DVFLAG    1-32      13-22
DVSHH1    1-59
DVSHH2    1-59
DVSHH3    1-59
DVSTAT    1-76
DX$NST    1-32      13-22
DZTXT     1-198
EDIT      1-73
EDITHD    4-13      5-4#
EDMALD    1-42      31-21
EDTFIL    1-184
EM$ACL    1-54
EM$CAP    1-44
EM$CIP    1-53      22-18
EM$CLB    1-54
EM$CLN    1-53
EM$CLX    1-36
EM$CNO    1-44
EM$CPO    1-44
EM$CSE    1-70
EM$HNI    1-68      13-16
EM$ICL    1-52
EM$IDR    1-42
EM$ILN    1-53      1-54
EM$IST    1-42
EM$IUN    1-53
EM$NAD    1-46      10-37     10-47
EM$NPD    1-53      10-34
EM$NPR    1-45
EM$NSF    1-53      26-25     27-26
EM$NSL    1-54      34-25
EM$NUK    1-86      28-51
EM$OPR    1-36      11-45
EM$PTA    1-36      20-41
EM$PTU    1-36      20-43
EM$SLT    1-54      18-46     34-29
EM$SLW    1-54      34-59
EM$SPL    1-39      16-19
EM$STL    1-42      27-33
EM$TSL    1-54
EM$UIO    1-55      34-94
EM$WCO    1-40      18-96
EM$WC1    1-40      18-97
EM$WC2    1-40      18-98
EM$WC3    1-40      18-99
EM$WCM    1-40      18-11
EMTHD     4-14      5-173#
ERRHD     4-16      5-13#
ERRLOC    1-67      18-86
```

```
ERRSEV    1-134     23-5*
ESC       1-82
FC$CDX    1-163
FC$LNK    1-163
FD$NAM    1-163
FF        2-9#
FILNAM    1-165     1-166     25-21     25-28     25-31     25-36     25-42     25-42     25-48     26-40     26-45     28-23
FKILL     1-165     7-44      8-56      8-58      10-34     10-37     10-47     11-36     11-45     13-33     13-39     14-39
          14-92     16-19     18-11     18-46     18-92     22-18     25-44     25-80     26-25     26-57     27-26     27-33
          27-50     28-51     28-55     28-59     29-43     29-54     30-17
FPRINT    1-165     15-38     20-41     20-43     21-8      21-18     22-8
FSTDL     1-89
FSTIOL    1-62      1-79
GAGMSG    1-190
GENTOP    1-103
GETKCH    1-77      12-8      12-11
GRT1      1-144
GTRD50    1-170     7-40      8-28      12-19     20-21
H.CSR     1-76
H.VEC     1-76
HANBSY    1-175     13-33
HANCHN    1-64      8-57      8-57      11-28     11-28     11-30     11-30     11-35     11-35     11-41     11-41     11-44
          11-44     12-97     12-97     13-5      13-5      13-6      13-6      13-38     13-38
HANENT    1-75
HANPAR    1-76
HANSAV    6-4#      11-41     12-75     12-97
HANSIZ    1-75
HAZEL     1-77
HAZLFL    1-77
HAZLNO    1-77
HIMAP     1-142
HIPRI     1-176     21-18
HNBUF     1-174     11-27*    11-28     13-11
HUPARG    1-195     13-26
II$$SZ    1-46
II$FLG    1-46
II$NAM    1-46
II$NPV    1-48
II$PRV    1-48
IIBUF     1-46
ILLCMD    1-171
ILSW2     1-83
IN$ACT    1-157
IN$CMD    1-157
IN$CNT    1-157
INDACT    1-167
INDERR    1-106
INDHD     4-20      5-61#
INDSAV    1-157     14-84
INDSTA    1-106
INFOMT    1-171
INGADR    1-46
INGEMT    1-46
INSTBL    1-46
INSTBN    1-47
INVDAT    1-200
```

```
INVDEV    1-199
INVEC     1-158
INVLDM    1-195
INVLDN    1-172
INVLDO    8-57#      12-26
INVOPT    1-165      1-171      1-186      8-58       34-108
INVSOP    8-55       8-58#
INVTIM    1-191
IOABFL    1-61       5-181      5-182
IOABHD    4-22       5-180#
ITRMTP    1-159
JCXPGS    1-142
JCXSMS    1-198
JPWDEV    1-39       20-25*
JPWFLG    1-39       20-19*     20-37*     20-52*     20-57*     20-62*     20-67*     20-72*     20-77*     20-82*     20-87*     20-92*
          20-97*
JPWTYP    1-39       20-36*
JSTKND    1-102
JSWLOC    1-67
K52       1-73
KBMSG     1-180
KBTX      1-184
KCSIBF    1-169
KCSIMS    1-170
KDOCIN    1-26
KED       1-73
KILEMT    1-192
KL3CLR    1-87
KL4CLR    1-127
KMNBAS    1-154      29-16
KMNCHN    1-94       12-96      29-30      29-35
KMNHI     1-78       29-17*
KMNNAM    1-196      29-6
KMNPGS    1-79       29-24*
KMNSTK    1-79       29-25*
KMNSTR    1-79       29-26*
KMNTOP    1-79       29-15*
KMONHD    4-23       5-31#
KMPRMT    1-138      15-10
L         4-9        4-9#       4-10       4-10#      4-11       4-11#      4-12       4-12#      4-13       4-13#      4-14       4-14#
          4-15       4-15#      4-16       4-16#      4-17       4-17#      4-18       4-18#      4-19       4-19#      4-20       4-20#
          4-21       4-21#      4-22       4-22#      4-23       4-23#      4-24       4-24#      4-25       4-25#      4-26       4-26#
          4-27       4-27#      4-28       4-28#      4-29       4-29#      4-30       4-30#      4-31       4-31#      4-32       4-32#
          4-33       4-33#      4-34       4-34#      4-35       4-35#      4-36       4-36#      4-37       4-37#      4-38       4-38#
          4-39       4-39#      4-40       4-40#      4-41       4-41#      4-42       4-42#      4-43       4-43#      4-44       4-44#
          4-45       4-45#      4-46       4-46#      4-47       4-47#      4-48       4-48#      4-49       4-49#      4-50       4-50#
          4-51       4-51#      4-52       4-52#      4-53       4-53#      4-54       4-54#      4-55       4-55#      4-56       4-56#
          4-57       4-57#      4-58       4-58#      4-59       4-59#      4-60       4-60#      4-61       4-61#      4-62       4-62#
          4-63       4-63#      5-5        5-5#       5-6        5-6#       5-7        5-7#       5-8        5-8#       5-14       5-14#
          5-15       5-15#      5-16       5-16#      5-17       5-17#      5-18       5-18#      5-19       5-19#      5-25       5-25#
          5-26       5-26#      5-32       5-32#      5-33       5-33#      5-34       5-34#      5-35       5-35#      5-36       5-36#
          5-37       5-37#      5-38       5-38#      5-39       5-39#      5-45       5-45#      5-46       5-46#      5-47       5-47#
          5-53       5-53#      5-54       5-54#      5-55       5-55#      5-56       5-56#      5-62       5-62#      5-63       5-63#
          5-69       5-69#      5-70       5-70#      5-71       5-71#      5-72       5-72#      5-73       5-73#      5-79       5-79#
          5-80       5-80#      5-81       5-81#      5-82       5-82#      5-83       5-83#      5-84       5-84#      5-85       5-85#
          5-86       5-86#      5-87       5-87#      5-88       5-88#      5-89       5-89#      5-90       5-90#      5-91       5-91#
```

	5-92	5-92#	5-93	5-93#	5-94	5-94#	5-95	5-95#	5-96	5-96#	5-97	5-97#
	5-98	5-98#	5-99	5-99#	5-105	5-105#	5-106	5-106#	5-112	5-112#	5-113	5-113#
	5-114	5-114#	5-115	5-115#	5-116	5-116#	5-117	5-117#	5-118	5-118#	5-119	5-119#
	5-120	5-120#	5-121	5-121#	5-122	5-122#	5-123	5-123#	5-129	5-129#	5-130	5-130#
	5-131	5-131#	5-132	5-132#	5-133	5-133#	5-134	5-134#	5-135	5-135#	5-136	5-136#
	5-137	5-137#	5-138	5-138#	5-139	5-139#	5-140	5-140#	5-141	5-141#	5-142	5-142#
	5-148	5-148#	5-149	5-149#	5-150	5-150#	5-151	5-151#	5-152	5-152#	5-153	5-153#
	5-154	5-154#	5-155	5-155#	5-156	5-156#	5-157	5-157#	5-158	5-158#	5-159	5-159#
	5-160	5-160#	5-161	5-161#	5-167	5-167#	5-168	5-168#	5-174	5-174#	5-175	5-175#
	5-181	5-181#	5-182	5-182#	5-188	5-188#	5-189	5-189#	5-190	5-190#	5-191	5-191#
	5-192	5-192#	5-193	5-193#	5-194	5-194#	5-195	5-195#	5-196	5-196#	5-197	5-197#
	5-198	5-198#	5-199	5-199#	5-200	5-200#	5-201	5-201#	5-202	5-202#	5-203	5-203#
	5-204	5-204#	5-205	5-205#	5-206	5-206#	5-212	5-212#	5-213	5-213#	5-214	5-214#
	5-215	5-215#	5-216	5-216#	5-217	5-217#	5-218	5-218#	5-219	5-219#	5-225	5-225#
	5-231	5-231#	5-237	5-237#	5-238	5-238#						
LA120	1-147											
LA12FL	1-148											
LA12NO	1-148											
LA36	1-147											
LA36FL	1-126											
LA36NO	1-126											
LACTIV	1-80											
LAFSIZ	1-83											
LANGHD	4-24	5-166#										
LBSPRI	1-160	22-21										
LCBIT	1-147											
LCDTYP	1-66											
LCLTXT	1-57											
LCLUNT	1-62											
LCOL	1-82	1-133										
LCONTM	1-91											
LCPUHI	1-91											
LCPULO	1-91											
LD$RON	1-121	30-30	30-35									
LDBASE	1-122	25-59										
LDCLEN	1-71	30-25	30-41									
LDDEVX	1-124	25-51										
LDDMT	30-40	32-7#										
LDFLAG	1-122	30-30*	30-35*									
LDMNT	1-70											
LDNAM	1-169											
LDNAME	1-122	32-21*	33-15									
LDOPHD	1-177	5-68#	30-14									
LDPDEV	1-122	25-58	32-19*									
LDSIZE	1-122											
LF	2-4#											
LF$IN	1-161	24-51	24-56	24-62	25-68							
LF$OPN	1-162	24-21	25-68									
LF$OUT	1-161	24-50	24-57	24-62	25-68							
LF$WRT	1-162	24-40	24-45	25-68								
LFWLIM	1-83											
LICTXT	1-144											
LINBUF	1-78											
LINCNT	1-80											
LINCUR	1-83											
LINFRE	1-191											

```
LINIR     1-65
LINNXT    1-78
LINPNT    1-80
LINRTS    1-65
LITIME    1-105
LJSW      1-102
LMXLN     1-158
LMXNUM    1-156
LMXPRM    1-159
LNAME     1-60
LNBLKS    1-107
LNMAP     1-113
LNPRIM    1-113
LNSBLK    1-108
LNSPAC    1-117
LOCKTX    1-180
LOFSPC    1-110    26-46
LOGASN    1-172    25-22    26-41    28-25
LOGBAS    1-132    1-134    25-55*   25-59*
LOGBLK    1-161    24-24*   25-70*
LOGBUF    1-161    24-23    25-69
LOGCHK    1-133    33-27
LOGCHN    1-161    24-25    25-42    25-42
LOGCLS    1-176    24-15    25-6
LOGDEA   31-11    32-8     33-8#
LOGDVU    1-132    1-134    25-53*   25-54*   25-58*
LOGFLG    1-161    24-21    24-40*   24-45*   24-50*   24-51*   24-56*   24-57*   24-62*   25-68*
LOGPTR    1-161    24-23*   25-69*
LOMAP     1-142
LOTBUF    1-81
LOTNXT    1-81
LOTPNT    1-81
LOTSIZ    1-82
LOTSPC    1-82
LOUTIR    1-65
LP$7BT    1-52
LP$ODD    1-52
LP$PAR    1-52
LP$SPD    1-52
LPRG1     1-145
LPRG2     1-145
LPRI      1-160
LPROG     1-90
LPROJ     1-90
LRBFIL    1-114
LRDTIM    1-80
LSCCA     1-110
LSECPT    1-95
LSPEMT    6-27#
LSTACT    1-78
LSTATE    1-140
LSTDL     1-89
LSTHL     1-62
LSTIOL    1-62
LSTMX     1-156
LSTPL     1-135
```

LSTPRM	1-131										
LSTSL	1-140										
LSTSPL	1-76										
LSUCF	1-86										
LSW	1-61										
LSW11	1-32	20-103*	20-109*								
LSW2	1-104										
LSW2S	1-109										
LSW3	1-109	17-6*									
LSW4	1-127										
LSW5	1-87	5-25	5-26	5-32	5-33	5-45	5-46	14-43*	14-48*	14-49*	14-55*	14-56*
	14-86*	14-96*									
LSW6	1-155	5-167	5-168	5-174	5-175	10-48*					
LSW7	1-159	5-34	5-53	5-54	5-55	5-56	5-62	5-63	14-61*	14-62*	14-67*	14-68*
	14-73*	14-74*	14-79*	28-45*	34-31*	34-32	34-41*	34-47*	34-53*	34-68*	34-74*	34-80*
	34-87*										
LSW8	1-123	5-188	5-189	5-190	5-191	5-192	5-193	5-194	5-195	5-196	5-197	5-198
	5-199	5-200	5-201	5-202	5-203	5-204	5-205	5-206			
LSW9	1-85	5-38	5-39	10-38*	10-40*	17-7*	26-21	27-22			
LTRMTP	1-148	18-44	34-27								
LTSCMD	1-117										
LUNAME	1-90										
MAXALC	1-63										
MAXASN	1-106										
MAXAVL	1-174	9-28	10-16								
MAXMEM	1-67										
MAXMTX	1-186										
MAXPRI	1-67	1-160	21-6	22-6	22-12						
MAXSEC	1-95										
MAXWEM	18-87	18-100#									
MDT	1-77										
MHNSIZ	1-96										
MHNSMS	1-97										
MINTIM	1-95										
MISSEQ	1-175	14-39	28-55								
MNBASE	1-185										
MNBPC	1-184										
MNFLGS	1-184										
MNTARG	1-195										
MNTDEV	1-170	33-23*									
MNTFUL	1-172										
MNTOP	1-185										
MNTTXT	1-197										
MONAR1	1-185										
MONAR2	1-185										
MONHD	1-185										
MONVEC	1-160	14-87	14-90	14-97	14-100	23-9					
MSGBUF	1-190										
MSGEND	1-190										
MTOPHD	1-171										
MUL32	1-200										
MXCSR	1-156										
MXDTR	1-156										
MXJADR	1-69										
MXJMEM	1-68										
MXJPRI	1-160	21-12	21-16	21-19	21-22	22-16	22-19*				

MXPRMT	1-138	15-36								
MXVEC	1-158									
NAMTOP	1-144									
NARGS	4-8#	4-9	4-9	4-10	4-10	4-11	4-11	4-12	4-12	4-13	4-13	4-14
	4-14	4-15	4-15	4-16	4-16	4-17	4-17	4-18	4-18	4-19	4-19	4-20
	4-20	4-21	4-21	4-22	4-22	4-23	4-23	4-24	4-24	4-25	4-25	4-26
	4-26	4-27	4-27	4-28	4-28	4-29	4-29	4-30	4-30	4-31	4-31	4-32
	4-32	4-33	4-33	4-34	4-34	4-35	4-35	4-36	4-36	4-37	4-37	4-38
	4-38	4-39	4-39	4-40	4-40	4-41	4-41	4-42	4-42	4-43	4-43	4-44
	4-44	4-45	4-45	4-46	4-46	4-47	4-47	4-48	4-48	4-49	4-49	4-50
	4-50	4-51	4-51	4-52	4-52	4-53	4-53	4-54	4-54	4-55	4-55	4-56
	4-56	4-57	4-57	4-58	4-58	4-59	4-59	4-60	4-60	4-61	4-61	4-62
	4-62	4-63	4-63	5-4#	5-5	5-5	5-6	5-6	5-7	5-7	5-8	5-8
	5-13#	5-14	5-14	5-15	5-15	5-16	5-16	5-17	5-17	5-18	5-18	5-19
	5-19	5-24#	5-25	5-25	5-26	5-26	5-31#	5-32	5-32	5-33	5-33	5-34
	5-34	5-35	5-35	5-36	5-36	5-37	5-37	5-38	5-38	5-39	5-39	5-44#
	5-45	5-45	5-46	5-46	5-47	5-47	5-52#	5-53	5-53	5-54	5-54	5-55
	5-55	5-56	5-56	5-61#	5-62	5-62	5-63	5-63	5-68#	5-69	5-69	5-70
	5-70	5-71	5-71	5-72	5-72	5-73	5-73	5-78#	5-79	5-79	5-80	5-80
	5-81	5-81	5-82	5-82	5-83	5-83	5-84	5-84	5-85	5-85	5-86	5-86
	5-87	5-87	5-88	5-88	5-89	5-89	5-90	5-90	5-91	5-91	5-92	5-92
	5-93	5-93	5-94	5-94	5-95	5-95	5-96	5-96	5-97	5-97	5-98	5-98
	5-99	5-99	5-104#	5-105	5-105	5-106	5-106	5-111#	5-112	5-112	5-113	5-113
	5-114	5-114	5-115	5-115	5-116	5-116	5-117	5-117	5-118	5-118	5-119	5-119
	5-120	5-120	5-121	5-121	5-122	5-122	5-123	5-123	5-128#	5-129	5-129	5-130
	5-130	5-131	5-131	5-132	5-132	5-133	5-133	5-134	5-134	5-135	5-135	5-136
	5-136	5-137	5-137	5-138	5-138	5-139	5-139	5-140	5-140	5-141	5-141	5-142
	5-142	5-147#	5-148	5-148	5-149	5-149	5-150	5-150	5-151	5-151	5-152	5-152
	5-153	5-153	5-154	5-154	5-155	5-155	5-156	5-156	5-157	5-157	5-158	5-158
	5-159	5-159	5-160	5-160	5-161	5-161	5-166#	5-167	5-167	5-168	5-168	5-173#
	5-174	5-174	5-175	5-175	5-180#	5-181	5-181	5-182	5-182	5-187#	5-188	5-188
	5-189	5-189	5-190	5-190	5-191	5-191	5-192	5-192	5-193	5-193	5-194	5-194
	5-195	5-195	5-196	5-196	5-197	5-197	5-198	5-198	5-199	5-199	5-200	5-200
	5-201	5-201	5-202	5-202	5-203	5-203	5-204	5-204	5-205	5-205	5-206	5-206
	5-211#	5-212	5-212	5-213	5-213	5-214	5-214	5-215	5-215	5-216	5-216	5-217
	5-217	5-218	5-218	5-219	5-219	5-224#	5-225	5-225	5-230#	5-231	5-231	5-236#
	5-237	5-237	5-238	5-238							
NATXT	1-178									
NEDCHR	1-65									
NEWJSW	1-102									
NFRESB	1-129									
NINTXT	1-178									
NLINES	1-115									
NOCCL	1-177	29-54								
NOCIN	1-169									
NODAT	1-183									
NOEQL	14-29	14-39#								
NOFIL	1-181	1-189								
NOFLAG	1-171	7-8*	12-4*	12-14*	12-41					
NOFRDL	1-200									
NOIND	1-175	14-92								
NOKMON	1-196	29-43								
NOLDMT	1-183									
NONEMS	1-183									
NOOPTT	1-189									
NOPMGN	1-185									

```
NOPRG    1-165      1-169
NOSTRT   1-168
NOTAVL   1-184
NOTON    1-190
NOTXT    1-178
NOUDC    1-182
NSPLDV   1-121
NSWPMS   1-186
NUCHN    1-114
NUMDCD   1-138       4-28
NUMDEV   1-151      13-10
NUMON    1-83
NXTOPT  11-43       12-4#       12-109
OCTFIX   1-178
OCTPRT   1-198
ODTBAS   1-154
OF$$SZ   1-153
OF$DEV   1-152
OF$FIL   1-152
OF$FLG   1-152
OF$UNT   1-152
OFFEMT   1-192
OKFEND   1-96
OKFILE   1-96
OPRTXT   1-57
OPTLST   1-43
OT$RON   1-153
OTHRON   1-194
OTRMNT   1-196
OVRCOR   1-168
PO$DBG   1-47       10-35       10-45
PO$NAM   1-45
PO$OPR   1-45       11-42
PO$SPV   1-43
PO$SYS   1-34       26-23       27-24
P2$CGR   1-40       18-9
P2$TRM   1-45
PA$BEL   1-36       20-62       20-67
PA$BLD   1-35        5-156       5-157       5-158       5-159       5-160       5-161
PA$DSC   1-35        5-157       5-158
PA$DTS   1-34       20-82       20-87
PA$DWD   1-35        5-155       5-156       5-157       5-158       5-161
PA$FLG   1-34       20-19       20-72      20-77
PA$GRC   1-35        5-155       5-156       5-157       5-158       5-161
PA$HQL   1-35        5-155       5-156       5-157       5-158
PA$LET   1-35       20-52       20-57
PA$NWD   1-36       20-92       20-97
PA$UKC   1-35        5-155       5-156       5-157       5-158
PA$ULN   1-35        5-155       5-156       5-157       5-158       5-159       5-160       5-161
PASLIN   1-134
PAUMSG   1-164
PBFEND   1-118
PEKADR   1-38
PEKEMT   1-38
PEKSIZ   1-38
PF$IOW   1-155
```

```
PF$SYS    1-155
PFCO      1-43
PFSO      1-43
PHYMEM    1-101
PMBUSY    1-185
PNAME     1-151      1-174      13-12
POPCF     1-175
PRGALL    1-27
PRGSIZ    1-78
PRGTOP    1-78
PRIEMT    6-38#      21-23      22-23
PRIVAO    1-44
PRIVCO    1-47       10-35      10-45      11-42      26-23      27-24
PRIVC2    1-45       18-9
PRIVSO    1-43
PRMBUF    1-131
PRMEND    1-131
PRMPNT    1-130
PROSLT    1-52
PRSBEL    5-129      20-62#
PRSDEV    5-131      20-18#
PRSDRF    5-132      20-57#
PRSFLG    5-133      20-72#
PRSHD     5-128#     20-9
PRSKEY    5-135      20-102#
PRSLET    5-137      20-52#
PRSNBL    5-130      20-67#
PRSNFL    5-134      20-77#
PRSNKY    5-136      20-108#
PRSNST    5-139      20-87#
PRSNWD    5-142      20-97#
PRSSTM    5-138      20-82#
PRSTYP    5-140      20-31#
PRSWID    5-141      20-92#
PRTBUF    1-183
PRTDAT    1-199
PRTDC2    1-180
PRTDC3    1-180
PRTDEC    1-174      9-29       10-17      21-10      21-13      21-20      22-10      22-13      34-61
PRTFIX    1-177
PRTFNM    1-183
PRTLN     1-179
PRTPCT    1-193
PRTR50    1-199
PRTSPC    1-177
PRTTIM    1-199
PRTTMD    1-182
PRTTMV    1-181
PRTTOD    1-199
PRTTTP    1-178
PRTUNM    1-179
PRTWRN    1-203      13-16      13-24      34-25      34-29      34-59      34-94      34-108
PRVCLR    14-14#
PRVLST    1-47
PRVOPT    1-43
PRVSET    14-6#
```

```
PUSHCF    1-166
PVNPW     1-43
PVON      1-87
PWTHD     5-147#    20-33
QHDMS1    1-59
QHDMS2    1-59
QUME      1-147
QUMEFL    1-149
QUMENO    1-149
R$CHN     1-41      24-27     24-31
R$XCHN    1-41      24-31
R50BUF    1-170     7-39*     7-41      7-42      7-48*     7-67      8-29      11-6      11-11*    11-27     12-27     12-29
          20-25
R50C10    6-6#
R50CLO    6-5#
R50COM    1-110     26-33
R50DIR    1-166
R50DK     1-199
R50DSK    1-172
R50DUP    1-167
R50IND    1-166
R50K52    1-167
R50KED    1-167
R50KEX    1-167
R50KMN    1-37
R50LD     1-172     7-52      30-6
R50LDO    1-170     30-9      33-22
R50LD7    1-172     7-54
R50LOG    1-176     25-14
R50MON    1-200
R50NO     1-173     8-29
R50PIP    1-167
R50SAV    1-166     28-19
R50SY     1-166
R50TSX    1-167
R50TT     1-195
R50UCL    1-167
R50VIR    1-168
RC$$SZ    1-51
RC$AEP    1-50
RC$AGE    1-50
RC$CNT    1-49
RC$EXC    1-50
RC$FLG    1-50
RC$GBL    1-50
RC$LCG    1-52
RC$LEN    1-51
RC$NAM    1-50
RC$OWN    1-49
RC$PVT    1-51
RC$USE    1-50
RCBBAS    1-51
RCBEND    1-51
RCLHD     5-104#    34-105
RCLREV    1-34      34-109*
RDB       1-139
```

RDBEND	1-139											
RDCMD	1-164	4-17	4-58	9-40	10-28	10-39	10-41	10-49	13-34	14-10	14-18	14-24
	14-37	14-44	14-50	14-57	14-63	14-69	14-75	14-80	14-91	14-101	15-50	16-30
	17-8	18-82	20-14	21-26	22-26	23-10	23-19	23-26	24-11	26-53	27-13	28-46
	29-53	30-26	30-31	30-36	30-42	31-25	34-17	34-110				
RDERM	1-168											
REMTXT	1-57											
RESDEV	1-153											
RJEMT	1-37											
RNMS	1-179											
RONTXT	1-184											
RSR	1-156											
RSTIND	5-33	14-96#										
RSTPRV	1-44											
RT$$SZ	1-139											
RT$NAM	1-139											
RUNDEV	1-107											
RUNEMT	1-168											
RUNHD	1-164											
RUNMS	1-200											
S$HICP	1-92											
S$INWT	1-93											
S$IOFN	1-88											
S$IOWT	1-145											
S$LOW	1-90											
S$MSWT	1-94											
S$OTFN	1-88											
S$OTLO	1-88											
S$OTWT	1-93											
S$QMIO	1-86											
S$QUSR	1-145											
S$RT	1-90											
S$SFWT	1-93	1-145										
S$SPCB	1-146											
S$SPDB	1-146											
S$SPND	1-87											
S$TMWT	1-93											
S$TTFN	1-88											
S$TTSC	1-89											
S$TWFN	1-88											
S9600	1-93											
SBPFIL	5-231	27-17#										
SBPHD	5-230#	27-11										
SBPSUF	1-36	27-38										
SC$ERR	1-33	5-15										
SC$FTL	1-33	5-17										
SC$NON	1-33	5-19										
SC$SEV	1-33	5-16										
SC$SUC	1-33											
SC$UNC	1-33	5-18										
SC$WRN	1-33	5-14										
SCHAIN	1-124											
SCNOPS	1-37	18-22	20-10	24-10	27-12	34-13						
SD$BAK	1-127											
SD$DEL	1-119											
SD$FLK	1-120											

```
SD$HLD    1-130
SD$SNG    1-129
SD$WFM    1-120
SDBLK     1-121
SDBU      1-127
SDBUF1    1-121
SDCB      1-135
SDCBND    1-135
SDCBSZ    1-140
SDDVU     1-139
SDFHD     1-132
SDFLAG    1-120
SDFORM    1-120
SDNAME    1-140
SDSFCB    1-119
SDSKIP    1-127
SEARCH    1-164     7-27      8-43      20-34     26-11     30-15     34-106
SEEDIT    5-5       14-43#
SEK52     5-7       14-54#
SEKED     5-6       14-55#
SERFLG    1-61      32-12*    32-15*
SETBIT    5-25      5-38      5-45      5-62      5-167     5-188     5-190     5-192     5-194     5-196     5-198     5-200
          5-202     5-204     14-7#
SETBYT    14-28#
SETCKP    11-31     11-41#
SETCL     1-32      7-70
SETCNW    5-47      29-49#
SETDEV    7-76      11-6#
SETECO    5-8       14-48#
SETEMT    5-174     10-45#
SETESU    4-15      17-6#
SETHD     1-173     4-8#      7-26
SETHST    1-32      4-19
SETIND    5-32      14-84#
SETIVD    11-15     11-17     11-29     11-35#
SETIVS    12-45     12-64     12-99     13-38#
SETKNW    5-37      29-5#
SETLD     7-58      30-5#
SETLOF    4-26      26-6#
SETLOG    4-25      24-5#
SETMPR    4-27      22-5#
SETNDC    4-28      9-44#
SETNSH    4-46      23-24#
SETOPT    12-23#
SETPRC    1-34      4-32
SETPRI    4-34      21-5#
SETPRS    4-35      4-36      4-63      20-5#
SETPRT    4-33      15-5#
SETPRV    4-22      8-14#
SETREC    4-44      34-105#
SETSBP    4-49      27-7#
SETSEV    5-14      5-15      5-16      5-17      5-18      5-19      23-5#
SETSHT    4-45      23-16#
SETSL     4-48      34-8#
SETSUB    4-10      4-12      4-13      4-14      4-16      4-20      4-23      4-24      4-47      4-57      4-61      8-15#
SETSVB    4-31      9-9#
```

SETSVC	9-10	9-12#										
SETSVL	4-11	4-18	4-21	4-29	4-30	4-37	4-38	4-39	4-40	4-41	4-42	4-43
	4-54	4-55	4-56	9-11#								
SETSVV	9-22#	9-50										
SETSYP	4-50	4-51	16-5#									
SETTTY	1-34	4-52	4-53									
SETUKM	5-35	5-36	28-7#									
SETUP	12-105	13-5#										
SETWIN	4-62	18-5#										
SETWRD	1-42											
SF$1ST	1-130											
SF$BSY	1-129											
SF$HLD	1-130											
SFFILE	1-146											
SFFLAG	1-132											
SFFORM	1-129											
SFID	1-76											
SFNMBL	1-129											
SFQLNK	1-132											
SFUSER	1-146											
SH$$SZ	1-74	12-31										
SH$FLG	1-74	12-43	12-48	12-56								
SH$NAM	1-74	12-27	12-29									
SH$RTN	1-74	12-36										
SH$VAL	1-74	12-24	12-84									
SHMTH1	1-182											
SHMTH2	1-182											
SHOHD	1-197											
SHRRCB	1-51											
SHRRCN	1-51											
SHTMSG	1-192											
SIGHD	4-47	5-187#										
SIZEMT	1-195											
SIZVAL	1-169											
SJEMT	1-37											
SKPSPC	1-26	7-6	8-20	8-35	12-5	12-15	12-50	12-54	12-103	15-9	19-27	25-7
	26-29	28-11										
SLDCLN	5-69	30-25#										
SLDEMP	5-70	31-5#										
SLDFRE	5-71	30-40#										
SLDNWR	5-73	30-35#										
SLDWRT	5-72	30-30#										
SLEHD	5-78#	34-12										
SLKDOF	1-55	34-40	34-88									
SLKDON	1-55	34-34	34-81									
SLMXLN	1-203	34-60										
SLOKED	5-80	5-81	5-82	34-80#								
SLOLET	5-86	5-92	34-68#									
SLONLT	5-87	5-93	34-74#									
SLONOP	5-79	5-85	5-94	5-98	5-99	34-100#						
SLONTT	5-96	34-53#										
SLOOFF	5-88	34-40#										
SLOON	5-89	5-90	34-23#									
SLORT	5-83	5-91	34-87#									
SLOTTY	5-95	34-47#										
SLOUNI	5-84	34-94#										

```
SLOWID    5-97      34-59#
SMRSIZ    1-143
SO$NO     1-75      12-43
SO$NVL    1-75      12-48
SO$OCT    1-75      12-56
SOPALC    1-66
SOPDAT    1-66
SOPTIM    1-66
SPACE1    1-49
SPACE2    1-179
SPACE3    1-179
SPACE5    1-180
SPACE6    1-183
SPACTV    1-187     1-188
SPCF      1-189
SPDTX1    1-178
SPFLK     1-189
SPFUL     1-189
SPGEMT    1-189
SPLACT    1-192
SPLCHN    1-92
SPLHD     1-186
SPLHLA    1-177     25-63
SPLPND    1-194
SPSNG     1-187     1-189
SPUBUF    1-69
SPWFM     1-187     1-188
SRTSIZ    1-143
SRTSMS    1-194
SRTTXT    1-197
SSRMAP    1-197
STCDOF    5-238     10-40#
STCDON    5-237     10-32#
STCHNB    4-9       10-4#
STLGAL    5-212     24-62#
STLGCL    5-214     24-15#
STLGCN    1-27      5-213     24-20#
STLGFL    5-215     25-4#
STLGHD    1-176     5-211#    24-9
STLGIN    5-216     24-50#
STLGNW    5-219     24-45#
STLGOT    5-217     24-56#
STLGWR    5-218     24-40#
STLOFL    5-225     26-21#
STLOHD    5-224#    26-10
STNOVR    1-34      4-60
STPASK    1-194
STPFLG    1-92      23-17*    23-25*
STRWRD    5-181     5-182     14-22#
STUCLF    5-53      14-61#
STUCLL    5-55      14-73#
STUCLM    5-54      14-67#
STUCLN    5-56      14-79#
STVRFY    1-34      4-59
SUBARO    1-184
SUBTXT    1-197
```

```
SUCS      1-144
SUM1      1-193
SUM2      1-193
SUM3      1-193
SUM4      1-193
SUM5      1-194
SUM6      1-194
SUM7      1-194
SUMS      1-144
SUPCOD    1-144
SWPTX     1-180
SXBPNT    1-69
SYASHD    1-59
SYHD1     1-179
SYHD2     1-179
SYINDX    1-151
SYNAME    1-152    28-31
SYPSWD    1-39     16-24
SYSAV     1-164
SYSDAT    1-141
SYTIMH    1-141
SYTIML    1-141
SYUNIT    1-151
TAB       2-8#
TALEMT    1-114
TBLOVF    1-173
TC132     6-21#    18-60
TC80      6-22#    18-66
TCNORM    6-20#    18-78
TCREV     6-19#    18-72
TECO      1-73
TK1SEC    1-143
TK1VAL    1-141
TM$AUT    1-57
TM$CDS    1-56
TM$CEN    1-56
TM$CLO    1-58
TM$CL1    1-58
TM$CL2    1-58
TM$CL3    1-58
TM$CL4    1-58
TM$CL5    1-58
TM$CL6    1-58
TM$CNG    1-56
TM$GBL    1-49
TM$HPE    1-56
TM$HPR    1-55
TM$IN1    1-38
TM$IN2    1-38
TM$LCL    1-49
TM$LPR    1-55
TM$MRS    1-32     13-24
TM$NAD    1-60
TM$NNR    1-38
TM$NSD    1-60
TM$PR1    1-55
```

```
TM$PR2    1-55
TM$PVA    1-44
TM$PVC    1-44
TM$RD1    1-49
TM$RD2    1-49
TM$SDN    1-60
TM$XBK    1-37
TMIDLH    1-71
TMIOH     1-71
TMIOWH    1-70
TMSWPH    1-71
TMSWTH    1-71
TMTOTH    1-70      1-193
TMTOTL    1-70      1-193
TMUSRH    1-70
TOTMMS    1-197
TOTON     1-92
TOTXT     1-176     21-11      22-11
TRGRET    1-144
TRMHD1    1-56
TRMHD2    1-56
TRMSTR    1-166
TSKST1    1-5#      1-26
TSR       1-156
TSXLN     1-144
TSXSIT    1-144
TSXSMS    1-198
TXTCL     1-91
UC$MDC    1-163
UC$NDC    1-163
UCHAN     1-112
UCIDEF    1-64      28-30
UCISPC    1-96      28-36      28-42
UCLBLK    1-162
UCLCMD    1-26
UCLDAT    1-162
UCLHD     4-57      5-52#
UCLNAM    1-117
UERSEV    1-134
UFORM     1-104
UFPTRP    1-119
UHIMEM    1-107
UKMNAM    1-85
UMSSMS    1-197
UMSYTP    1-89
UPTMMS    1-192
USPLCH    1-92
USRMMS    1-198
USRSTK    1-68
USTART    1-103
UTRPAD    1-67
VCORTM    1-138     4-11
VCSHNB    1-95      10-22*
VDBFLG    1-82      10-32
VHIPCT    1-136     4-18
VIMAGE    1-102
```

```
VINTIO   1-128     4-21
VLDSYS   1-86      7-56
VNUMDC   1-138     9-49
VOFFTM   1-205     4-29
VONTM    1-205     4-30
VPRIDF   1-42      22-15
VQUAN0   1-135     1-136     4-37
VQUAN1   1-136     4-38
VQUAN2   1-136     4-42
VQUAN3   1-135     1-136     4-43
VQUN1A   1-136     4-39
VQUN1B   1-128     4-40
VQUN1C   1-128     4-41
VSLEDT   1-203     34-23
VT100    1-147     18-44     34-27
VT10FL   1-149
VT10NO   1-149
VT200    1-146     18-44     34-27
VT2007   1-146
VT2008   1-146
VT20FL   1-150
VT20NO   1-150
VT52     1-147     18-44     34-27
VT52FL   1-148
VT52NO   1-126
VTMIN    1-205     4-54
VTMLOC   1-205     4-55
VTMOUT   1-205     4-56
VUSPHN   1-205     4-31
WEM      18-91     18-96#    18-100
WF132    5-123     6-14#     18-38     18-58
WF80     5-116     6-15#     18-64
WFDEL    5-114     5-118     6-11#     18-30
WFNORM   5-113     5-117     6-13#     18-76
WFREV    5-115     5-120     6-12#     18-70
WILDFL   1-72
WILDHD   4-61      5-24#
WINBIT   5-113     5-114     5-115     5-116     5-117     5-118     5-119     5-120     5-123     19-56#
WINCOL   5-112     19-4#
WINDEL   6-55#     18-32
WINFLG   6-7#      18-17*    18-26     19-56*
WINHD    5-111#    18-21
WINMAK   6-43#     18-15*    18-16*    18-40*    18-47     19-12*    19-40*    19-50*
WINMAP   6-50#     18-53
WINNOS   5-122     19-50#
WINSCR   5-121     19-23#
WLDNAM   2-12#
XAREA    1-165     11-28     11-30     11-41     12-75     12-96     12-97     13-5      14-87     14-90     14-97     14-100
         23-9      25-42     28-42     29-6      29-11     29-30     29-35     29-50     29-52
YESTXT   1-178
ZCLR     1-156
```

```
...CM0   25-28
...CM1   11-28    11-30    11-41    12-75    12-96    12-97    13-5     25-42    28-42    29-6     29-11    29-30
         29-35    29-50    29-52
...CM2   11-28    11-28    11-30    11-30    11-30    11-30    11-41    12-75    12-96    12-97    13-5     13-5
         13-5     13-5     14-87    14-90    14-90    14-97    14-100   14-100   23-9     23-9     25-42    25-42
         25-42    28-42    28-42    29-6     29-6     29-11    29-11    29-11    29-11    29-30    29-35    29-50
         29-50    29-52
...CM3   8-57     11-35    11-44    12-74    12-95    13-6     13-38    28-44    29-34
...CM5   9-28     9-30     10-16    10-18    11-28    11-30    11-41    12-75    12-96    12-97    13-5     14-87
         14-90    14-97    14-100   18-60    18-66    18-72    18-78    21-11    21-14    21-21    22-11    22-14
         23-9     25-28    25-42    28-42    29-6     29-11    29-30    29-35    29-50    29-52    31-21    34-34
         34-40    34-62    34-81    34-88
...CM6   14-87    14-90    14-97    14-100   23-9
...CM7   11-30    13-5     29-11
.CLOSE   1-20#    13-6     28-44
.CSIGE   1-18#
.CSISP   1-16#
.DATE    1-19#
.DSTAT   1-22#    25-28
.ENTER   1-21#    25-42
.EXIT    1-21#    29-39
.FPROT   1-22#
.GTIM    1-19#
.GTLIN   1-19#
.GVAL    1-22#    14-87    14-97
.HERR    1-22#
.LOOKU   1-20#    11-28    28-42    29-6     29-50
.PRINT   1-20#    9-28     9-30     10-16    10-18    18-60    18-66    18-72    18-78    21-11    21-14    21-21
         22-11    22-14    31-21    34-34    34-40    34-62    34-81    34-88
.PURGE   1-17#    8-57     11-35    11-44    12-74    12-95    13-38    29-34
.PVAL    1-22#    14-90    14-100   23-9
.READW   1-17#    11-30    29-11
.REOPE   1-18#    12-75    12-96    12-97    29-35
.SAVES   1-18#    11-41    29-30    29-52
.SERR    1-22#
.SPFUN   1-19#
.SRESE   1-16#
.TTOUT   1-16#
.TTYIN   1-17#
.TTYOU   1-17#
.WRITW   1-21#    13-5
CMDDEF   3-47#    4-9      4-10     4-11     4-12     4-13     4-14     4-15     4-16     4-17     4-18     4-19
         4-20     4-21     4-22     4-23     4-24     4-25     4-26     4-27     4-28     4-29     4-30     4-31
         4-32     4-33     4-34     4-35     4-36     4-37     4-38     4-39     4-40     4-41     4-42     4-43
         4-44     4-45     4-46     4-47     4-48     4-49     4-50     4-51     4-52     4-53     4-54     4-55
         4-56     4-57     4-58     4-59     4-60     4-61     4-62     4-63     5-5      5-6      5-7      5-8
         5-14     5-15     5-16     5-17     5-18     5-19     5-25     5-26     5-32     5-33     5-34     5-35
         5-36     5-37     5-38     5-39     5-45     5-46     5-47     5-53     5-54     5-55     5-56     5-62
         5-63     5-69     5-70     5-71     5-72     5-73     5-79     5-80     5-81     5-82     5-83     5-84
         5-85     5-86     5-87     5-88     5-89     5-90     5-91     5-92     5-93     5-94     5-95     5-96
         5-97     5-98     5-99     5-105    5-106    5-112    5-113    5-114    5-115    5-116    5-117    5-118
         5-119    5-120    5-121    5-122    5-123    5-129    5-130    5-131    5-132    5-133    5-134    5-135
         5-136    5-137    5-138    5-139    5-140    5-141    5-142    5-148    5-149    5-150    5-151    5-152
         5-153    5-154    5-155    5-156    5-157    5-158    5-159    5-160    5-161    5-167    5-168    5-174
         5-175    5-181    5-182    5-188    5-189    5-190    5-191    5-192    5-193    5-194    5-195    5-196
         5-197    5-198    5-199    5-200    5-201    5-202    5-203    5-204    5-205    5-206    5-212    5-213
```

	5-214	5-215	5-216	5-217	5-218	5-219	5-225	5-231	5-237	5-238		
FABORT	3-15#	7-44	8-56	8-58	10-34	10-37	10-47	11-36	11-45	13-33	13-39	14-39
	14-92	16-19	18-11	18-46	18-92	22-18	25-44	25-80	26-25	26-57	27-26	27-33
	27-50	28-51	28-55	28-59	29-43	29-54	30-17					
FERR	3-4#	15-38	20-41	20-43	21-8	21-18	22-8					
FWARN	3-23#	13-16	13-24	34-25	34-29	34-59	34-94	34-108				
TBLDEF	3-35#	4-8	5-4	5-13	5-24	5-31	5-44	5-52	5-61	5-68	5-78	5-104
	5-111	5-128	5-147	5-166	5-173	5-180	5-187	5-211	5-224	5-230	5-236	
TBLEND	3-61#	4-64	5-9	5-20	5-27	5-40	5-48	5-57	5-64	5-74	5-100	5-107
	5-124	5-143	5-162	5-169	5-176	5-183	5-207	5-220	5-226	5-232	5-239	

```
TSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX                              -PlusTSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-Pl
usTSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-PlusT  S&H Computer Systems, Inc.  SX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-
PlusTSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-Plu                              sTSX-PlusTSX-PlusTSX-PlusTSX-PlusTSX-PlusTS
```

```
              TTTTT  SSS  K   K  SSS  TTTTT  222
                T     S   S K K S   S   T    2   2
                T     S     K K  S       T        2
                T     SSS   KK   SSS     T       2
                T       S K K    S   S   T      2
                T     S   S K K S   S    T     2
                T     SSS  K   K  SSS    T     22222
```

```
     TTTTTTTTTT   SSSSSS   KK       KK   SSSSSS   TTTTTTTTTT   222222
     TTTTTTTTTT   SSSSSS   KK       KK   SSSSSS   TTTTTTTTTT   222222
         TT     SS      SS KK      KK   SS     SS     TT       22       22
         TT     SS      SS KK     KK    SS     SS     TT       22         22
         TT     SS         KK   KK      SS            TT                 22
         TT     SS         KK   KK      SS            TT                22
         TT       SSSSSS   KKKK         SSSSSS        TT              22
         TT       SSSSSS   KKKK         SSSSSS        TT              22
         TT            SS  KK   KK           SS       TT            22
         TT            SS  KK   KK           SS       TT            22
         TT     SS     SS  KK    KK    SS    SS       TT          22
         TT     SS     SS  KK    KK    SS    SS       TT          22
         TT       SSSSSS   KK       KK   SSSSSS       TT          2222222222
         TT       SSSSSS   KK       KK   SSSSSS       TT          2222222222
```

```
     [[[    000   55555        000   333    ]]]
      [    0   0  5           0   0  3   3    ]
      [    0  00  5           0  00      3    ]
      [    0 0 0  5555        0 0 0     33    ]
      [    00  0      5   ,,  00  0      3    ]
      [    0   0  5   5   ,,  0   0  3   3    ]
     [[[    000   555   ,,     000   333    ]]]
```

```
     L    IIIII N   N EEEEE                    1      4
     L      I   NN  N E                         11     44
     L      I   N N N E           =====        1 1    4 4
     L      I   N  NN EEEE                      1     44444
     L      I   N   N E           =====        1      4
```