TSKST2 -- Keyboard SET Command MACRO V05.05 Thursday 19-Jan-89 14:58
Table of contents

```
                                        .TITLE  TSKST2 -- Keyboard SET Command routines
                                        .ENABL  LC
                                        .DSABL  GBL
000000                                  .CSECT  TSKST2
000000                          TSKST2:
                                ;
                                ;  TSKST2 is the portion of TSKMON that contains the code
                                ;  to implement the SET command.
                                ;
                                ;  Copyright 1978, 1979, 1980, 1981, 1982, 1983, 1984, 1985, 1986, 1987, 1988.
                                ;  S&H Computer Systems, Inc.
                                ;  Nashville, Tennessee
                                ;
                                ;  Macro calls
                                ;
                                        .MCALL  .CSISPC,.TTOUTR,.SRESET
                                        .MCALL  .READW,.TTYIN,.TTYOUT,.PURGE
                                        .MCALL  .CSIGEN,.SAVEST,.REOPEN
                                        .MCALL  .GTLIN,.GTIM,.DATE,.SPFUN
                                        .MCALL  .PRINT,.CLOSE,.LOOKUP
                                        .MCALL  .WRITW,.ENTER,.EXIT
                                        .MCALL  .SERR,.HERR,.FPROT,.GVAL,.PVAL
                                ;
                                ;  Global definitions
                                ;
                                        .GLOBL  SETCL,SETHST,SVT227,SVT228
                                        .GLOBL  SETPRC,SETTTY,STVRFY,STNOVR,SVT50,SVT100,SVT200
                                        .GLOBL  TSKST2,CMDHD,CMDOFF,KDOCIN,SKPSPC,UCLCMD
                                        .GLOBL  DORUN,CMDFRM,CMDDSN,STLGCN,DATTIM,PRGALL
                                        .GLOBL  DLCEMT,ALCDEV,CMDSHO,CMDSET,CMDWHO,CMDMEM,CMDUSE
                                ;
                                ;  Global references
                                ;
                                        .GLOBL  EM$LAS,EM$LFD,EM$CLU,LWINDO,CLSFRS,LCXTBL,MXTTCT,EM$TMT
                                        .GLOBL  PA$GRC,PA$UKC,PA$DSC,PA$BLD,PA$ULN,PA$DWD,PA$HQL,PA$LET
                                        .GLOBL  EM$PTA,EM$PTU,SBPSUF,CHKCLU,EM$CLX,PA$BEL,EM$OPR,PA$NWD
                                        .GLOBL  SCNOPS,TM$XBK,CL$XLN,CKCLUS,ACRSTR,R50KMN,SJEMT,RJEMT
                                        .GLOBL  PEKEMT,PEKADR,PEKSIZ,TM$NNR,CDBUF,CDGET,TM$IN1,TM$IN2
                                        .GLOBL  SYPSWD,EM$SPL,ACRTXT,JPWDEV,JPWTYP,JPWFLG,EM$BYP
                                        .GLOBL  EM$WC0,EM$WC1,EM$WC2,EM$WC3,EM$WCM,P2$CGR
                                        .GLOBL  CXTRMN,R$CHN,R$XCHN,CHNSIZ,C.USED,CL$EPN,CL$EPS,CLEOFS
                                        .GLOBL  EM$STL,EM$IST,CLSFEP,EM$IDR,VPRIDF,SETWRD
                                        .GLOBL  CLRPRV,OPTLST,PFSO,PFCO,PVNPW,PO$SPV,PRIVSO,PRVOPT
                                        .GLOBL  TM$PVA,TM$PVC,PRIVAO,EM$CNO,EM$CPO,EM$CAP,RSTPRV
                                        .GLOBL  CHKEQ,CKACOJ,PO$OPR,CKSYPV,P2$TRM,PRIVC2,PO$NAM,EM$NPR
                                        .GLOBL  INSTBL,INGADR,INGEMT,IIBUF,II$NAM,II$FLG,II$$SZ,EM$NAD
                                        .GLOBL  INSTBN,AF$SCA,AF$NOW,AF$MEM,PO$DBG,PRIVCO,PRVLST
                                        .GLOBL  ABRTAD,ABRTCD,CINFLG,$VNOTT,II$PRV,II$NPV,PO$BYP
                                        .GLOBL  TM$RD1,TM$RD2,TM$LCL,TM$GBL,SPACE1,RC$OWN,RC$CNT
                                        .GLOBL  RC$EXC,RC$AGE,RC$AEP,RC$USE,RC$FLG,RC$GBL,RC$NAM
                                        .GLOBL  RC$LEN,RC$PVT,RCBBAS,RCBEND,RC$$SZ,SHRRCB,SHRRCN
                                        .GLOBL  LP$SPD,LP$PAR,LP$ODD,LP$7BT,EM$ICL,PROSLT,RC$LCG
                                        .GLOBL  EM$NPD,EM$ILN,EM$CIP,EM$NSF,EM$IUN,EM$CLN,EM$IVN
                                        .GLOBL  EM$ILN,EM$ACL,EM$TSL,EM$CLB,EM$NSL,EM$SLT,EM$SLW
                                        .GLOBL  SLKDON,SLKDOF,EM$UIO,TM$PR1,TM$PR2,TM$LPR,TM$HPR
                                        .GLOBL  TM$HPE,TM$CNG,TM$CDS,TM$CEN,TRMHD1,TRMHD2,AT$DEV
                                        .GLOBL  OPRTXT,CLLINE,LCLTXT,REMTXT,TM$AUT,CLFREE,CLUNIT,CLVERS
```

```
                .GLOBL  TM$CL0, TM$CL1, TM$CL2, TM$CL3, TM$CL4, TM$CL5, TM$CL6
                .GLOBL  QHDMS1, QHDMS2, DVSHH1, DVSHH2, DVSHH3, SYASHD, DKASHD
                .GLOBL  TM$NAD, ALCHD1, ALCHD2, TM$NSD, TM$SDN, LNAME
                .GLOBL  CORUSR, LSW, $CTRLO, SERFLG, IOABFL, $CHACT, $STSNG
                .GLOBL  LSTHL, LCLUNT, FSTIOL, LSTIOL, CL$LIX, CW$PRO, CONFG2
                .GLOBL  CL$RQH, CL$WQH, MAXALC, ALCTBL, ALCEND
                .GLOBL  AD$DVU, AD$JOB, AD$$SZ, UCIDEF, HANCHN
                .GLOBL  NEDCHR, LOUTIR, LINIR, LINRTS, CLOTIR
                .GLOBL  CO$DEF, CL$COL, LCDTYP, SOPALC, SOPDAT, SOPTIM
                .GLOBL  UTRPAD, JSWLOC, ERRLOC, MAXMEM, MAXPRI
                .GLOBL  USRSTK, $KINIT, CFSTK, MXJMEM, DFJMEM, EM$HNI
                .GLOBL  SPUBUF, SXBPNT, MXJADR, CLSFCH
                .GLOBL  TMTOTH, TMTOTL, TMUSRH, TMIOWH, LDMNT, EM$CSE
                .GLOBL  TMSWTH, TMIDLH, TMIOH, TMSWPH, LDCLEN
                .GLOBL  WILDFL, $NOIN, $NOWTT, $HITTY
                .GLOBL  TECO, EDIT, KED, K52, $1STLG, $DIBOL
                .GLOBL  SH$VAL, SH$NAM, SH$$SZ, SH$RTN, SH$FLG
                .GLOBL  SO$NVL, SO$OCT, SO$NO, HANENT, HANSIZ
                .GLOBL  H.CSR, H.VEC, DVSTAT, SFID, ACRSPD, HANPAR, LSTSPL
                .GLOBL  HAZEL, HAZLFL, HAZLNO, $MLOCK, MDT, GETKCH
                .GLOBL  LINBUF, LINNXT, LSTACT, PRGTOP, PRGSIZ, KMNHI
                .GLOBL  KMNTOP, KMNPGS, KMNSTK, KMNSTR, CXTPAG, FSTIOL
                .GLOBL  LINPNT, LINCNT, LACTIV, LRDTIM, CS$RON
                .GLOBL  LOTBUF, LOTNXT, LOTPNT, $VTESC
                .GLOBL  LOTSIZ, LOTSPC, LCOL, $SLKED, ESC, VDBFLG
                .GLOBL  LAFSIZ, LFWLIM, LINCUR, NUMON, ILSW2
                .GLOBL  $DBKMN
                .GLOBL  $CARUP, DOASGN, UKMNAM, $UKMON, LSW9
                .GLOBL  LSUCF, $CCLRN, VLDSYS, EM$NUK, S$QMIO
                .GLOBL  KL3CLR, $PRGLK, LSW5, PVON, S$SPND, $AUTO
                .GLOBL  S$TWFN, S$TTFN, S$OTFN, S$IOFN, S$OTLO
                .GLOBL  LSTDL, FSTDL, $DETCH, UMSYTP, S$TTSC
                .GLOBL  $DISCN, LPROJ, LPROG, LUNAME, S$RT, S$LOW
                .GLOBL  LCPUHI, LCPULO, LCONTM, $CTRLS, $SPLJB, TXTCL
                .GLOBL  STPFLG, TOTON, USPLCH, SPLCHN, S$HICP
                .GLOBL  S$INWT, S$OTWT, S$TMWT, S$SFWT, S9600
                .GLOBL  S$MSWT, CFBUF, CFEND, CCLSAV, KMNCHN
                .GLOBL  MINTIM, LSECPT, MAXSEC, $EMTTR, VCSHNB
                .GLOBL  OKFILE, OKFEND, $CLTST, UCISPC, MHNSIZ
                .GLOBL  CASTBR, CASCBR, CASTBW, CASCUP, MHNSMS
                .GLOBL  CASTRO, CASTWO, CLTOTL, CO$DTR, CLSFSP
                .GLOBL  CO$CR, CO$FF, CO$FFO, CO$LC, CO$TAB, CO$CTL
                .GLOBL  CO$LFI, CO$LFO, CO$BNI, CO$BNO, CL$OPT
                .GLOBL  CL$LEN, CL$SKP, CL$WID, CL$LIN, PHYMEM
                .GLOBL  LJSW, CTRLTT, NEWJSW, JSTKND, VIMAGE
                .GLOBL  USTART, GENTOP, BOTDEV, BOTUNI, CSHALC
                .GLOBL  $CTRLC, LSW2, $INKMN, CHAIN, UFORM
                .GLOBL  $SGQ0, $SGQ3, LITIME
                .GLOBL  MAXASN, $CFABT, INDSTA, INDERR
                .GLOBL  RUNDEV, LNBLKS, CXTBAS, CXTWDS, UHIMEM
                .GLOBL  $DILUP, CSHDEV, CSHDVN, LNSBLK
                .GLOBL  LSW3, LSW2S, $DUPRN
                .GLOBL  $FORM, $TAB, LSCCA, $CFSOT, LOFSPC, R50COM
                .GLOBL  $PAGE, $SCOPE, $ECHO, $LC, $8BIT, CHKALC, $ALTER
                .GLOBL  UCHAN, $FORMO, $CFALL, $CFDCC, $CFCCL
                .GLOBL  LNPRIM, LNMAP, CW$50H, CONFIG, $SUCF
                .GLOBL  $DOOFF, NUCHN, LRBFIL, CFIND, TALEMT
```

```
                .GLOBL  C.CSW,C.DEVQ,C.SBLK,NLINES,CO$BBT
                .GLOBL  CD$NAM,CD$DVU,CD$BAS,CD$JOB,CD$$SZ,CD$$UB
                .GLOBL  LTSCMD,LNSPAC,CFNEST,UCLNAM
                .GLOBL  $CFOPN,CFSEND,PBFEND,CFSP,$TTGAG
                .GLOBL  UFPTRP,SDSFCB,SD$DEL,CFLFL4,$UCLCF
                .GLOBL  SDFLAG,SD$FLK,SD$WFM,SDFORM,$UCLRN
                .GLOBL  SDBUF1,SDBLK,NSPLDV,LD$RON,$UCLCM,$UCLCL
                .GLOBL  LDNAME,LDSIZE,LDFLAG,LDBASE,LDPDEV
                .GLOBL  LSW8,$SGQ1,$SGQ1A,$SGQ1B,$SGQ1C,$SGQ2,$SGIIO,$SGHIO
                .GLOBL  $DEFER,CFCHAN,SCHAIN,LDDEVX,$SGALL
                .GLOBL  CFPNT,CFBLK,$QUIET,DIABFL
                .GLOBL  DIABNO,VT52NO,LA36NO,LA36FL
                .GLOBL  LSW4,KL4CLR,SDSKIP,SDBU,SD$BAK
                .GLOBL  $INCOR,$KED,VQUN1B,VINTIO,VQUN1C
                .GLOBL  SF$BSY,SFFORM,SD$SNG,SFNMBL,NFRESB
                .GLOBL  SD$HLD,SF$HLD,CURPRM,PRMPNT,SF$1ST
                .GLOBL  LSTPRM,PRMBUF,PRMEND,CFSPND
                .GLOBL  SDFHD,SFFLAG,SFQLNK,CFHOLD,LOGDVU,LOGBAS
                .GLOBL  LCOL,$QTSET,$TECO,CD$TOP,LOGCHK
                .GLOBL  $WILD,ERRSEV,VERSEV,PASLIN,LOGBAS,LOGDVU
                .GLOBL  LSTPL,SDCB,SDCBND,VQUAN0,VQUAN3
                .GLOBL  VQUAN1,VQUN1A,VQUAN2,VHIPCT,VQUAN0,VQUAN3
                .GLOBL  DCTRD,DCCRD,DCTWR,DCCWR,ASNSRC
                .GLOBL  VCORTM,NUMDCD,VNUMDC,KMPRMT,MXPRMT
                .GLOBL  RDB,RDBEND,RT$NAM,RT$$SZ,CLDEVX,SDDVU
                .GLOBL  SDNAME,SDCBSZ,LSTSL,LSTATE
                .GLOBL  TK1VAL,CINDAT,SYSDAT,SYTIMH,SYTIML
                .GLOBL  BASMAP,LOMAP,HIMAP,JCXPGS
                .GLOBL  SMRSIZ,SRTSIZ,CSHSIZ,TK1SEC
                .GLOBL  TSXLN,TSXSIT,GRT1,TRGRET,LICTXT,SUPCOD,NAMTOP,SUMS,SUCS
                .GLOBL  LPRG1,LPRG2,S$QUSR,S$IOWT,S$SFWT
                .GLOBL  S$SPDB,S$SPCB,SFUSER,SFFILE,VT200,VT2007,VT2008
                .GLOBL  LCBIT,LA36,LA120,VT52,VT100,DIABLO,QUME
                .GLOBL  ADM3A,LTRMTP,LA12FL,LA12NO,VT52FL
                .GLOBL  VT10FL,VT10NO,QUMEFL,QUMENO,ADM3FL
                .GLOBL  VT20FL,VT20NO
                .GLOBL  ADM3NO,SYINDX,SYUNIT,NUMDEV,PNAME
                .GLOBL  OF$DEV,OF$UNT,OF$FIL,OF$FLG,SYNAME
                .GLOBL  OF$$SZ,OT$RON,RESDEV,$TAPE
                .GLOBL  KMNBAS,ODTBAS,$CTRLD
                .GLOBL  LSW6,$SNWTT,PF$SYS,PF$IOW,$DEBUG
                .GLOBL  RSR,TSR,LMXNUM,LSTMX,MXDTR,ZCLR,MXCSR
                .GLOBL  $INDDF,$INDRN,IN$ACT,IN$CNT,IN$CMD,INDSAV
                .GLOBL  $PHONE,INVEC,LMXLN,MXVEC,$INIT,$DEAD,$HARD
                .GLOBL  ITRMTP,LMXPRM,LSW7,$INDAB,CFSTS,CF$IND,CF$QUT
                .GLOBL  CFABLV,MONVEC,LBSPRI,MAXPRI,MXJPRI,LPRI,$SYSPS
                .GLOBL  LOGCHN,LOGFLG,LOGPTR,LOGBUF,LOGBLK
                .GLOBL  LF$OPN,LF$WRT,UCLBLK,UCLDAT
                .GLOBL  CSHHD,FC$CDX,FC$LNK,FD$NAM,UC$NDC,UC$MDC,CVTUC
                .GLOBL  CMDBUF,PAUMSG,RDCMD,DKSAV,SYSAV,CVTTAB,RUNHD,SEARCH
                .GLOBL  INVOPT,FKILL,ABRTCF,ACRFN,XAREA,FILNAM,NOPRG,FPRINT
                .GLOBL  PUSHCF,TRMSTR,FILNAM,R50DIR,R50SY,R50IND,R50SAV
                .GLOBL  INDACT,R50DUP,R50PIP,R50KED,R50K52,R50KEX,R50TSX,R50UCL
                .GLOBL  BLKO,RDERM,R50VIR,NOSTRT,RUNEMT,OVRCOR
                .GLOBL  BADSAV,LDNAM,NOPRG,NOCIN,SIZVAL,ASKLNM,BADCMD,KCSIBF
                .GLOBL  ASDEX,KCSIMS,ASNOVF,GTRD50,R50BUF,R50LDO,MNTDEV,DMTARG
                .GLOBL  DEADEV,CHKMNT,CHKMTX,INFOMT,NOFLAG,MTOPHD,INVOPT,ILLCMD
```

```
                .GLOBL  R50LD, INVLDN, R50DSK, ACRFIL, BDFNAM, LOGASN, MNTFUL, R50LD7
                .GLOBL  TBLOVF, SETHD, CSIMS2, CKPRIV, R50NO, AMBOPT, ACRDEC
                .GLOBL  MAXAVL, PRTDEC, DEVUNT, PNAME, HNBUF, CKTERM
                .GLOBL  ACROCT, HANBSY, CSIMS1, MISSEQ, NOIND, POPCF
                .GLOBL  BADPMT, BADPRI, TOTXT, CRLF, HIPRI, STLGHD, LOGCLS, R50LOG
                .GLOBL  BDLGOP, SPLHLA, NOCCL, LDOPHD, PRTFIX, PRTSPC
                .GLOBL  DLTXT, OCTFIX, PRTTTP, NATXT, SPDTX1, NOTXT, YESTXT, NINTXT
                .GLOBL  PRTUNM, SYHD1, SYHD2, PRTLN, SPACE2, DETTXT, SPACE3, RNMS
                .GLOBL  SWPTX, LOCKTX, SPACE5, PRTDC3, KBMSG, DIVIDE, PRTDC2
                .GLOBL  COLOO, CPUAH, CPUAL, PRTTMV, NOFIL, CMDBUF, CALUCL
                .GLOBL  NOUDC, DEVHD1, ASNHD1, ASNHD2, SHMTH1, SHMTH2, PRTTMD
                .GLOBL  CVDVNM, SPACE6, PRTBUF, PRTFNM, NONEMS, NODAT, NOLDMT
                .GLOBL  SUBARG, EDTFIL, RONTXT, NOTAVL, KBTX, MNFLGS, MNBPC
                .GLOBL  DELSPC, MNBASE, MNTOP, MONHD, MONAR1, NOPMGN, PMBUSY, MONAR2
                .GLOBL  NSWPMS, MAXMTX, CURMTX, CHKDLM, SPLHD, AMBOPT, INVOPT
                .GLOBL  DEVIDL, COAL, ALDEX, COAD, SPACTV, SPWFM, DEVIDL, SPSNG
                .GLOBL  COAL, ALDEX, ALDBLK, COAD, SPACTV, SPWFM, DEVIDL
                .GLOBL  SPSNG, SPFUL, SPCF, SPFLK, NOFIL, SPGEMT, NOOPTT
                .GLOBL  BDLIN, MSGBUF, MSGEND, NOTON, GAGMSG
                .GLOBL  LINFRE, DJABMS, DLMSG, INVTIM, DMTALL
                .GLOBL  SHTMSG, AUTHFN, SPLACT, DOSTOP, OFFEMT, KILEMT, UPTMMS
                .GLOBL  TMTOTH, DIVSOR, TMTOTL, PRTPCT, SUM1, SUM2, SUM3, SUM4
                .GLOBL  SUM5, SUM6, SUM7, OTHRON, SPLPND, STPASK, SRTSMS, CHKTTD
                .GLOBL  SIZEMT, ASNOVF, INVLDM, CSIMS4, MNTARG, HUPARG, R50TT
                .GLOBL  KMNNAM, NOKMON, CCLNAM, OTRMNT, CHKDEV, DMTSUB, CMDCCL
                .GLOBL  SHOHD, SUBTXT, MNTTXT, SRTTXT, TOTMMS, UMSSMS, SSRMAP
                .GLOBL  TSXSMS, USRMMS, JCXSMS, DZTXT, OCTPRT
                .GLOBL  PRTR50, PRTDAT, PRTTOD, PRTTIM, INVDEV, ALFN, R50DK
                .GLOBL  DETHD, DETARG, RUNMS, NOFRDL, R50MON, INVDAT, MUL32, COAF
                .GLOBL  AR$PRJ, AR$PRG, AR$CON, AR$CNT, AR$CPH, AR$CPL, AR$UNM
                .GLOBL  AR$DMY, AR$$SZ, ARNRPB, $SLON, $SLTTY, $SLLET
                .GLOBL  PRTWRN, SLMXLN, VSLEDT, $LOFCF, CSHMSG
                .GLOBL  AF$HIE, AF$NOI, $NOINT, AF$PLK, AF$DBG
                .GLOBL  AF$IOP, $RNIOP
```

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

```
                        ;-----------------------------------------------------
                        ;  Macro to cause a fatal error message to be printed.
                        ;
                              .MACRO  FERR    MSG
                              MOV     R5,-(SP)
                              MOV     MSG,R5
                              CALL    FPRINT
                              MOV     (SP)+,R5
                              .ENDM   FERR
                        ;
                        ;-----------------------------------------------------
                        ;  Macro to print a fatal error message, clean up
                        ;  and then jump to RDCMD.
                        ;
                              .MACRO  FABORT  MSG
                              MOV     MSG,R5
                              JMP     FKILL
                              .ENDM   FABORT
                        ;
                        ;-----------------------------------------------------------
                        ;  Macro to print a warning message
                        ;
                              .MACRO  FWARN   MSG
                              MOV     R5,-(SP)
                              MOV     MSG,R5
                              CALL    PRTWRN
                              MOV     (SP)+,R5
                              .ENDM   FWARN
                        ;
                        ;-----------------------------------------------------------
                        ;  Macro to start a standard option table.
                        ;  Name = 1 to 4 character table name.
                        ;  NA = Number of arguments per table entry.
                        ;
                              .MACRO  TBLDEF  NAME,NA
                        NARGS =       NA
                              .CSECT  CMDVS2
                        NAME'HD: .WORD  2*NA
                              .ENDM   TBLDEF
                        ;
                        ;-----------------------------------------------------------
                        ;  Macro to enter an option text name and a set of parameters
                        ;  into the currently open table.
                        ;  STRNG = Ascii name
                        ;  A,B,C = Set of option parameters to store in table with name.
                        ;
                              .MACRO  CMDDEF  STRNG,A,B,C
                              .CSECT  NAMES2
                        L     =       .
                              .ASCIZ  /STRNG/
                              .CSECT  CMDVS2
                              .WORD   L               ;POINTER TO NAME STRING
                              .WORD   A
                        .IIF  GE,<NARGS-2>    .WORD   B
                        .IIF  GE,<NARGS-3>    .WORD   C
                              .ENDM   CMDDEF
                        ;
```

```
                        ;-----------------------------------------------------------------
                        ;  Macro to end a set of table entries.
                        ;
                                .MACRO  TBLEND
                                .CSECT  CMDVS2
                                .WORD   0
                                .CSECT  TSKST2
                                .ENDM   TBLEND
```

```
                                        ;-------------------------------------------------------------------
                                        ;  Data areas
                                        ;
000000  125017  074773                  R50HST: .RAD50  /$HOST$/
000004  012256  000000  000000          CLDEV:  .RAD50  /CLN         /
000012  000000
000014  012276                          R50CLO: .RAD50  /CLO/
000016  013666                          R50C10: .RAD50  /C10/
000020  000000                          CPFLAG: .WORD   0
                                        ;
                                        ;  Flags stored in CPFLAG
                                        ;
                000001                  CPFSUS  =       1               ;Suspend process
                000002                  CPFRES  =       2               ;Resume process
                000004                  CPFAUT  =       4               ;Set authorized privileges
                                        ;
000022      000                         SETPRM: .BYTE   0               ;Indicates if SET is temp or perm
000023      000                         CPPID:  .BYTE   0               ;Process ID for SET PROCESS command
000024      000                         CPPRIO: .BYTE   0               ;Priority value for SET PROCESS command
000025                                  CPNAME: .BLKB   12.             ;Name buffer for SET PROCESS command
                                                .EVEN
                                        ;
                                        ;  Emt to assign a CL line to a time-sharing line
                                        ;
000042      000     155                 CLAEMT: .BYTE   0,155
000044  000000                                  .WORD   0               ;CL unit number
000046  000000                                  .WORD   0               ;Line number
                                        ;
                                        ;  Emt to clear XOFF status for a CL line (SET CLn XONBYPASS)
                                        ;
000050      001     155                 CLXEMT: .BYTE   1,155
000052  000000                                  .WORD   0               ;CL unit number
                                        ;
                                        ;  Emt to completely reset a CL line (SET CLn RESETBYPASS)
                                        ;
000054      002     155                 CLREMT: .BYTE   2,155
000056  000000                                  .WORD   0               ;CL unit number
                                        ;
                                        ;  Emt to cross connect our time-sharing line with a CL unit
                                        ;
000060      000     126                 TTXCL:  .BYTE   0,126
000062  000013                                  .WORD   13
000064  000000                                  .WORD   0
                                        ;
                                        ;  Emt to set line speed
                                        ;
000066      000     154                 LSPEMT: .BYTE   0,154
000070  000000                                  .WORD   0               ;Line number
000072  000000                                  .WORD   0               ;Speed code
                                        ;
                                        ;  Emt to reset XOFF status for a TT line
                                        ;
000074      001     154                 XONEMT: .BYTE   1,154
000076  000000                                  .WORD   0
                                        ;
                                        ;  EMT to set/reset DTR for a line
                                        ;
```

```
     57 000100     002     154          DTREMT: .BYTE   2,154           ;Assume raise DTR (3,154 = drop DTR)
     58 000102  000000                          .WORD   0
     59                                 ;
     60                                 ;  Emt to set privilege flags
     61                                 ;
     62 000104     001     150          PVSEMT: .BYTE   1,150
     63 000106     002     001                  .BYTE   2,1
     64 000110  000000G                         .WORD   PFSO
     65 000112  000000                          .WORD   0
     66                                 ;
     67                                 ;  Emt to clear privilege flags
     68                                 ;
     69 000114     001     150          PVCEMT: .BYTE   1,150
     70 000116     001     001                  .BYTE   1,1
     71 000120  000000G                         .WORD   PFCO
     72 000122  000000                          .WORD   0
     73                                 ;
     74                                 ;  Emt to start an inactive line
     75                                 ;
     76 000124     000     126          STAEMT: .BYTE   0,126
     77 000126  000020                          .WORD   20
     78 000130  000000                          .WORD   0
     79                                 ;
     80                                 ;  Emt to declare terminal type change to process windowing system
     81                                 ;
     82 000132     007     161          WINSTT: .BYTE   7,161
```

```
                        ;-----------------------------------------------------------------------
                        ;  Options for SET TT command
                        ;
000134                          TBLDEF  TT,3
000002                          CMDDEF  ADM*3A,SADM3A,0,0
000012                          CMDDEF  AL*TERNATE,SETTTB,LSW2,$ALTER
000022                          CMDDEF  AUTO*BAUD,SAUTO,0,0
000032                          CMDDEF  NOAUTO*BAUD,CLRTTP,LSW2,$AUTO
000042                          CMDDEF  BIT*S,STCLEN,0,0
000052                          CMDDEF  DEAD,SETDED,LSW3,$DEAD
000062                          CMDDEF  NODEAD,CLRTTP,LSW3,$DEAD
000072                          CMDDEF  DEC*WRITER,STLA36,0,0
000102                          CMDDEF  DEF*ER,SETTTB,LSW2,$DEFER
000112                          CMDDEF  NOD*EFER,CLRTTB,LSW2,$DEFER
000122                          CMDDEF  DI*ABLO,SDIAB,0,0
000132                          CMDDEF  DT*R,SETDTR,0,0
000142                          CMDDEF  NODT*R,CLRDTR,0,0
000152                          CMDDEF  E*CHO,SETTTB,LSW2,$ECHO
000162                          CMDDEF  NOE*CHO,CLRTTB,LSW2,$ECHO
000172                          CMDDEF  EIGHT*BIT,SETTTB,LSW2,$8BIT
000202                          CMDDEF  NOEIGHT*BIT,CLRTTB,LSW2,$8BIT
000212                          CMDDEF  FORM,SETTTB,LSW2,$FORM
000222                          CMDDEF  NOFORM,CLRTTB,LSW2,$FORM
000232                          CMDDEF  FORMO,SETTTB,LSW4,$FORMO
000242                          CMDDEF  NOFORMO,CLRTTB,LSW4,$FORMO
000252                          CMDDEF  G*AG,SETTTB,LSW7,$TTGAG
000262                          CMDDEF  NOG*AG,CLRTTB,LSW7,$TTGAG
000272                          CMDDEF  HAZEL*TINE,STHAZL,0,0
000302                          CMDDEF  HO*LD,RDCMD,0,0
000312                          CMDDEF  NOH*OLD,RDCMD,0,0
000322                          CMDDEF  LA1*20,SLA120,0,0
000332                          CMDDEF  LA3*6,STLA36,0,0
000342                          CMDDEF  LC,SETTTB,LSW2,$LC
000352                          CMDDEF  NOL*C,CLRTTB,LSW2,$LC
000362                          CMDDEF  LE*NGTH,RDCMD,0,0
000372                          CMDDEF  PAG*E,SETTTB,LSW2,$PAGE
000402                          CMDDEF  NOP*AGE,CLRTTB,LSW2,$PAGE
000412                          CMDDEF  PAR*ITY,SETPAR,0,0
000422                          CMDDEF  NOPAR*ITY,STNOPR,0,0
000432                          CMDDEF  PHO*NE,SETTTP,LSW2,$PHONE
000442                          CMDDEF  NOPHO*NE,CLRTTP,LSW2,$PHONE
000452                          CMDDEF  QUI*ET,SETQUT,0,0
000462                          CMDDEF  NOQ*UIET,RSTQUT,0,0
000472                          CMDDEF  QUM*E,SQUME,0,0
000502                          CMDDEF  SC*OPE,SETTTB,LSW2,$SCOPE
000512                          CMDDEF  NOSC*OPE,SETNSC,0,0
000522                          CMDDEF  SEVEN*BIT,CLRTTB,LSW2,$8BIT
000532                          CMDDEF  SI*NGLE,SETTTB,LSW6,$STSNG
000542                          CMDDEF  NOSI*NGLE,CLRTTB,LSW6,$STSNG
000552                          CMDDEF  SPE*ED,STTTSP,0,0
000562                          CMDDEF  STA*RT,STTTST,0,0
000572                          CMDDEF  STAN*DARD,CLRTTB,LSW2,$ALTER
000602                          CMDDEF  STD,CLRTTB,LSW2,$ALTER
000612                          CMDDEF  SYSP*ASSWORD,SETTTP,LSW2,$SYSPS
000622                          CMDDEF  NOSYSP*ASSWORD,CLRTTP,LSW2,$SYSPS
000632                          CMDDEF  SYSPS,SETTTP,LSW2,$SYSPS
000642                          CMDDEF  NOSYSPS,CLRTTP,LSW2,$SYSPS
```

```
                                        CMDDEF  SYSP*WD,SETTTP,LSW2,$SYSPS
                                        CMDDEF  NOSYSP*WD,CLRTTP,LSW2,$SYSPS
                                        CMDDEF  TAB,SETTTB,LSW2,$TAB
                                        CMDDEF  NOTAB,CLRTTB,LSW2,$TAB
                                        CMDDEF  TAP*E,SETTTB,LSW2,$TAPE
                                        CMDDEF  NOTAP*E,CLRTTB,LSW2,$TAPE
                                        CMDDEF  TR*ANSLATE,STTRNS,0,0
                                        CMDDEF  VT1*00,SVT100,24.,0
                                        CMDDEF  VT2*00,SVT200,24.,0
                                        CMDDEF  VT220,SVT200,24.,0
                                        CMDDEF  VT2207,SVT227,24.,0
                                        CMDDEF  VT2208,SVT228,24.,0
                                        CMDDEF  VT240,SVT200,24.,0
                                        CMDDEF  VT241,SVT200,24.,0
                                        CMDDEF  VT5*0,SVT50,12.,0
                                        CMDDEF  VT5*2,SVT50,24.,0
                                        CMDDEF  W*AIT,SETTW,0,0
                                        CMDDEF  NOW*AIT,SETTNW,0,0
                                        CMDDEF  XON,SETXON,0,0
                                        CMDDEF  8BIT,SETTTB,LSW2,$8BIT
                                        CMDDEF  NO8BIT,CLRTTB,LSW2,$8BIT
                                        CMDDEF  7BIT,CLRTTB,LSW2,$8BIT
                                        TBLEND
                                ;
                                ;  Define options for SET TT [n] PARITY= command
                                ;
                                        TBLDEF  PAR,3
                                        CMDDEF  E*VEN,STPAR,LP$PAR
                                        CMDDEF  O*DD,STPAR,LP$PAR!LP$ODD
                                        CMDDEF  N*ONE,STPAR,0
                                        TBLEND
                                ;
                                ;
                                ;  Define options for the SET CLn command
                                ;
                                        TBLDEF  CLOP,2
                                        CMDDEF  BIT*S,STCLBT,0
                                        CMDDEF  CR,STCLOP,CO$CR
                                        CMDDEF  NOCR,CLCLOP,CO$CR
                                        CMDDEF  DTR,STCLOP,CO$DTR
                                        CMDDEF  NODTR,CLCLOP,CO$DTR
                                        CMDDEF  ENDP*AGES,STCLVL,CL$EPN
                                        CMDDEF  ENDS*TRING,CLESTR,0
                                        CMDDEF  FORM,STCLOP,CO$FF
                                        CMDDEF  NOFORM,CLCLOP,CO$FF
                                        CMDDEF  FF,STCLOP,CO$FF
                                        CMDDEF  NOFF,CLCLOP,CO$FF
                                        CMDDEF  FORMO,STCLOP,CO$FFO
                                        CMDDEF  NOFORMO,CLCLOP,CO$FFO
                                        CMDDEF  GR*APH,STCLGR,0.
                                        CMDDEF  NOGR*APH,STCLGR,132.
                                        CMDDEF  LC,STCLOP,CO$LC
                                        CMDDEF  NOLC,CLCLOP,CO$LC
                                        CMDDEF  TAB,STCLOP,CO$TAB
                                        CMDDEF  NOTAB,CLCLOP,CO$TAB
                                        CMDDEF  CTRL,STCLOP,CO$CTL
                                        CMDDEF  NOCTRL,CLCLOP,CO$CTL
```

```
                                        CMDDEF  LEN*GTH,STCLVL,CL$LEN
                                        CMDDEF  LFIN,STCLOP,CO$LFI
                                        CMDDEF  NOLFIN,CLCLOP,CO$LFI
                                        CMDDEF  LFOUT,STCLOP,CO$LFO
                                        CMDDEF  NOLFOUT,CLCLOP,CO$LFO
                                        CMDDEF  LIN*E,STCLLN,0
                                        CMDDEF  BININ,STCLOP,CO$BNI
                                        CMDDEF  NOBININ,CLCLOP,CO$BNI
                                        CMDDEF  BINOUT,STCLOP,CO$BNO
                                        CMDDEF  NOBINOUT,CLCLOP,CO$BNO
                                        CMDDEF  8BIT,STCLOP,CO$8BT
                                        CMDDEF  EIGHT*BIT,STCLOP,CO$8BT
                                        CMDDEF  NO8*BIT,CLCLOP,CO$8BT
                                        CMDDEF  NOEIGHT*BIT,CLCLOP,CO$8BT
                                        CMDDEF  7BIT,CLCLOP,CO$8BT
                                        CMDDEF  PAR*ITY,STCLPO,CLPRHD
                                        CMDDEF  NOPAR*ITY,STCLNP
                                        CMDDEF  SEVEN*BIT,CLCLOP,CO$8BT
                                        CMDDEF  TOP,CLTOP,0
                                        CMDDEF  TR*ANSLATE,CLTRNS,0
                                        CMDDEF  RES*ET,STCLRS,0
                                        CMDDEF  SKI*P,STCLVL,CL$SKP
                                        CMDDEF  WID*TH,STCLVL,CL$WID
                                        CMDDEF  SP*EED,STCLSP,0
                                        CMDDEF  VER*SION,STCLVR,0
                                        CMDDEF  XON,STCLXN,0
                                ;
                                ;  These two forms do not require allocation of the CL device, they
                                ;  do their own separate checking for BYPASS privilege instead.
                                ;  Leave them at the end of the table.
                                CLNCKA:
                                        CMDDEF  XONB*YPASS,STCLXB,0
                                        CMDDEF  RESETB*YPASS,STCLRB,0
                                        TBLEND
                                ;
                                ;  Define options for SET CL PARITY= command
                                ;
                                        TBLDEF  CLPR,2
                                        CMDDEF  E*VEN,STCLPR,LP$PAR
                                        CMDDEF  O*DD,STCLPR,LP$PAR!LP$ODD
                                        CMDDEF  N*ONE,STCLPR,0
                                        TBLEND
                                ;
                                ;  Define options for the SET HOST command
                                ;
                                        TBLDEF  HOST,1
                                        CMDDEF  DTE,HSTDTE
                                        CMDDEF  CL,HSTDTE
                                        CMDDEF  PO*RT,HSTPRT
                                        TBLEND
                                ;
                                ;  Define options for SET PROCESS command
                                ;
                                        TBLDEF  CPOP,2
                                        CMDDEF  PRIV*ILEGES,PRVOPT,0
                                        CMDDEF  ID*ENTIFICATION,CPPPID,0
                                        CMDDEF  NAM*E,CPPNAM,0
                                        CMDDEF  PRIO*RITY,CPPPRI,0
```

```
    172 001742                                  CMDDEF  SUSP*END,CPPFLG,CPFSUS,0
    173 001750                                  CMDDEF  RES*UME,CPPFLG,CPFRES,0
    174 001756                                  CMDDEF  AU*THORIZED,CPPFLG,CPFAUT,0
    175 001764                                  TBLEND
```

```
                                        .SBTTL  SET commands
                                        .SBTTL  .       Process
                                ;-------------------------------------------------------------------------------
                                ;  Process the SET PROCESS command.
                                ;
000134                          SETPRC:
                                ;
                                ;  Initialize cells which will hold values parsed by the command
                                ;
000134  116767  0000000 177661          MOVB    CORUSR,CPPID    ;Set process ID=our job
000142  105067  177657                 CLRB    CPNAME          ;No name specified
000146  105067  177652                 CLRB    CPPRIO          ;No priority
000152  005067  177642                 CLR     CPFLAG          ;No flags
000156  004767  0000000                CALL    CLRPRV          ;No privileges
                                ;
                                ;  Do command parsing
                                ;
000162  012704  001710'                 MOV     #CPOPHD,R4      ;Point to option driver list
000166  004767  0000000                CALL    OPTLST          ;Parse the command
                                ;
                                ;  Now process each option that was accrued
                                ;
000172  004767  000366                  CALL    CPSNAM          ;See if process name was specified
000176  004767  000644                  CALL    CPSPRI          ;See if priority was specified
000202  004767  000464                  CALL    CPSPRV          ;See if privileges were specified
000206  004767  000676                  CALL    CPSFLG          ;See if flags were specified
                                ;
                                ;  Finished command
                                ;
000212  000167  0000000         9$:     JMP     RDCMD           ;Finished command
```

```
                                ;*******************************************************************************
                                ;  SET PROCESS/IDENTIFICATION=job-number
                                ;
000216  010146                  CPPPID: MOV     R1,-(SP)
                                ;
                                ;  Accrue the ID value
                                ;
000220  004767  000000G                 CALL    ACRDEC
                                ;
                                ;  Make sure the ID value is valid
                                ;
000224  006301                          ASL     R1              ;Convert to word table index
000226  001413                          BEQ     10$             ;Invalid if zero
000230  020127  000000G                 CMP     R1,#LSTSL       ;Is this a valid job #?
000234  101010                          BHI     10$             ;Br if not
000236  032761  000000G 000000G         BIT     #$DILUP,LSW(R1) ;Is line logged on?
000244  001410                          BEQ     11$             ;Br if not
000246  110167  177551                  MOVB    R1,CPPID        ;Save ID value
                                ;
                                ;  Finished
                                ;
000252  012601                          MOV     (SP)+,R1
000254  000207                          RETURN
                                ;
                                ;  Invalid ID
                                ;
000256                          10$:    FABORT  #EM$ILN         ;Invalid job numver
000266                          11$:    FABORT  #NOTON          ;Line not logged on
```

```
                        ;-------------------------------------------------------------------------------
                        ;  SET PROCESS/PRIORITY=value
                        ;
000276  010146         CPPPRI: MOV     R1,-(SP)
000300  010546                 MOV     R5,-(SP)
                        ;
                        ;  Accrue the priority value
                        ;
000302  004767  000000G         CALL    ACRDEC          ;Accrue priority value
                        ;
                        ;  See if the priority value is valid
                        ;
000306  005701                  TST     R1              ;Must be > 0
000310  003406                  BLE     10$             ;Br if invalid
000312  120167  000000G         CMPB    R1,MXJPRI       ;Is it within valid range for job?
000316  101003                  BHI     10$             ;Br if not
000320  110167  177500          MOVB    R1,CPPRIO       ;Save priority value
000324  000424                  BR      9$
                        ;
                        ;  Invalid priority value
                        ;
000326                  10$:    FERR    #BADPRI         ;Invalid priority value
000342  012705  000001          MOV     #1,R5           ;Minimum prio value
000346  004767  000000G         CALL    PRTDEC
000352                          .PRINT  #TOTXT          ;"to"
000360  116705  000000G         MOVB    MXJPRI,R5       ;Max possible
000364  004767  000000G         CALL    PRTDEC
000370                          .PRINT  #CRLF
                        ;
                        ;  Finished
                        ;
000376  012605          9$:     MOV     (SP)+,R5
000400  012601                  MOV     (SP)+,R1
000402  000207                  RETURN
```

```
                        ;-------------------------------------------------------------------------------
                        ;  SET PROCESS/NAME="string"
                        ;
000404  010246         CPPNAM: MOV     R2,-(SP)
000406  010446                 MOV     R4,-(SP)
                        ;
                        ;  Make sure equal sign follows keyword
                        ;
000410  004767  0000000          CALL    CHKEQ           ;Check for equal sign
000414  012702  000025'          MOV     #CPNAME,R2      ;Point to buffer for name
                        ;
                        ;  See if string is enclosed in quote marks
                        ;
000420  004767  0000000          CALL    SKPSPC          ;Skip over any spaces
000424  111300                   MOVB    (R3),R0         ;Get 1st char of string
000426  120027  000042           CMPB    R0,#42          ;Quote mark?
000432  001403                   BEQ     3$              ;Br if yes
000434  120027  000047           CMPB    R0,#47          ;Apostrophe?
000440  001014                   BNE     1$              ;Br if not
                        ;
                        ;  String is enclosed in quote marks
                        ;
000442  004767  0000000  3$:     CALL    ACRSTR          ;Accrue a quoted string
000446  005700                   TST     R0              ;Is string empty?
000450  001423                   BEQ     2$              ;If empty, go blank fill it.
000452  020027  000014           CMP     R0,#12.         ;Is string too long?
000456  101026                   BHI     10$             ;Br if too long
000460  012704  0000000          MOV     #BLK0,R4        ;Point to accrued string
000464  112422           4$:     MOVB    (R4)+,(R2)+     ;Move to string buffer
000466  077002                   SOB     R0,4$
000470  000413                   BR      2$              ;Go blank fill rest of buffer if needed
                        ;
                        ;  Name string is not quoted.
                        ;  Move name to buffer till we hit end of string
                        ;
000472  111300           1$:     MOVB    (R3),R0         ;Get next char from string
000474  001411                   BEQ     2$              ;Br if hit end of command
000476  120027  000057           CMPB    R0,#'/          ;Start of next qualifier?
000502  001406                   BEQ     2$              ;Br if yes
000504  005203                   INC     R3              ;Point to next character in command
000506  020227  000041'          CMP     R2,#CPNAME+12.  ;Buffer overflow?
000512  103010                   BHIS    10$             ;Br if yes
000514  110022                   MOVB    R0,(R2)+        ;Store char into name buffer
000516  000765                   BR      1$
                        ;
                        ;  Fill remainder of buffer with blanks
                        ;
000520  020227  000041'  2$:     CMP     R2,#CPNAME+12.  ;Filled buffer yet?
000524  103011                   BHIS    9$              ;Br if yes
000526  112722  000040           MOVB    #' ,(R2)+       ;Pad with blanks
000532  000772                   BR      2$
                        ;
                        ;  Error: string is too long
                        ;
000534                  10$:     FERR    #EM$STL         ;String is too long
                        ;
                        ;  Finished
```

```
                                        ;
000550  012604                  9$:     MOV     (SP)+,R4
000552  012602                          MOV     (SP)+,R2
000554  000207                          RETURN
                                        ;-------------------------------------------------------------------------------
                                        ;  Set some flag for a SET PROCESS qualifier
                                        ;
000556  051467  177236          CPPFLG: BIS     (R4),CPFLAG     ;Set specified flag bit
000562  000207                          RETURN
```

```
                                ;*****************************************************************************
                                ;  See if NAME qualifier was specified.
                                ;
000564  010146                  CPSNAM: MOV     R1,-(SP)
000566  010246                          MOV     R2,-(SP)
                                ;
                                ;  See if name qualifier was specified
                                ;
000570  105767  177231                  TSTB    CPNAME          ;Was name qualifier specified?
000574  001423                          BEQ     9$              ;Br if not
                                ;
                                ;  Make sure we are changing for our own process
                                ;
000576  116701  177221                  MOVB    CPPID,R1        ;Get job index
000602  120167  000000G                 CMPB    R1,CORUSR       ;Affecting our own process?
000606  001021                          BNE     10$             ;Br if not
000610  032767  000000G 000000G         BIT     #PO$NAM,PRIVCO  ;Are we authorized to change our name?
000616  001421                          BEQ     11$             ;Br if not
000620  070127  000006                  MUL     #6.,R1          ;Each job has 12 character name
000624  062701  000000G                 ADD     #LUNAME,R1      ;Point to name cell for our job
000630  012702  000025'                 MOV     #CPNAME,R2      ;Point to name buffer
000634  012700  000014                  MOV     #12.,R0         ;Get # bytes to move
000640  112221                  1$:     MOVB    (R2)+,(R1)+     ;Set name for job
000642  077002                          SOB     R0,1$
                                ;
                                ;  Finished
                                ;
000644  012602                  9$:     MOV     (SP)+,R2
000646  012601                          MOV     (SP)+,R1
000650  000207                          RETURN
                                ;
                                ;  Cannot change name of another job
                                ;
000652                          10$:    FABORT  #EM$CNO
000662                          11$:    FABORT  #EM$NPR
```

```
                                ;-------------------------------------------------------------------------
                                ;  Change privilege flags for our process.
                                ;
000672  010246                 CPSPRV: MOV     R2,-(SP)
000674  010346                         MOV     R3,-(SP)
000676  010446                         MOV     R4,-(SP)
                                ;
                                ;  See if any privilege changes were specified
                                ;
000700  012702  000000G                MOV     #PFS0,R2        ;Flags to set
000704  012703  000000G                MOV     #PFC0,R3        ;Flags to clear
000710  012700  000000G                MOV     #PVNPW,R0       ;# words to check
000714  005722                 1$:     TST     (R2)+           ;Any flags to set?
000716  001004                         BNE     2$              ;Br if yes
000720  005723                         TST     (R3)+           ;Any flags to clear?
000722  001002                         BNE     2$              ;Br if yes
000724  077005                         SOB     R0,1$           ;Keep checking
000726  000433                         BR      9$              ;No privilege changes requested
                                ;
                                ;  Make sure we are changing privileges for our job
                                ;
000730  126767  177067  000000G 2$:     CMPB    CPPID,CORUSR    ;Changing privilege for our job?
000736  001033                         BNE     10$             ;Br if not
                                ;
                                ;  Set correct EMT function code depending on whether we are doing a
                                ;  normal privilege change or are changing authorized privileges.
                                ;
000740  012704  000001                 MOV     #1,R4           ;Assume normal priv change
000744  032767  000004  177046         BIT     #CPFAUT,CPFLAG  ;Changing authorized privileges?
000752  001402                         BEQ     4$              ;Br if not
000754  012704  000002                 MOV     #2,R4           ;Code to change authorized privileges
000760  110467  177123         4$:     MOVB    R4,PVSEMT+3     ;Set code in EMT arg block
000764  110467  177127                 MOVB    R4,PVCEMT+3
                                ;
                                ;  Set any specified privileges
                                ;
000770  005004                         CLR     R4              ;No privilege error detected
000772  012700  000104'                MOV     #PVSEMT,R0      ;Point to EMT arg block
000776  104375                         EMT     375             ;Set privilege flags
001000  103001                         BCC     3$              ;Br if no authorization violation
001002  005204                         INC     R4              ;Remember we had authorization violation
                                ;
                                ;  Clear any specified privileges
                                ;
001004  012700  000114'         3$:     MOV     #PVCEMT,R0      ;Point to EMT arg block
001010  104375                         EMT     375             ;Clear privilege flags
001012  005704                         TST     R4              ;Did we have an authorization violation?
001014  001010                         BNE     11$             ;Br if yes
                                ;
                                ;  Finished
                                ;
001016  012604                 9$:     MOV     (SP)+,R4
001020  012603                         MOV     (SP)+,R3
001022  012602                         MOV     (SP)+,R2
001024  000207                         RETURN
                                ;
                                ;  Can't change privilege for another job
```

```
      58                                ;
      59 001026                         10$:    FABORT  #EM$CPO         ;Can't change privilege for another job
      60                                ;
      61                                ;  Not authorized to give yourself this privilege
      62                                ;
      63 001036                         11$:    FABORT  #EM$CAP         ;Can't authorize that privilege
```

```
                                ;*****************************************************************
                                ;  See if we need to change priority for job.
                                ;
001046  010246                 CPSPRI: MOV     R2,-(SP)
                                ;
                                ;  See if a priority change was requested
                                ;
001050  105767  176750                  TSTB    CPPRIO          ;Was a priority specified
001054  001413                          BEQ     9$              ;Br if not
001056  116702  176741                  MOVB    CPPID,R2        ;Get ID # of job we want to affect
001062  004767  000000G                 CALL    CKACOJ          ;Can we access that job?
001066  103406                          BCS     9$              ;Br if not
                                ;
                                ;  Change priority value
                                ;
001070  116762  176730  000000G          MOVB    CPPRIO,LBSPRI(R2) ;Set base priority
001076  116762  176722  000000G          MOVB    CPPRIO,LPRI(R2) ;Set running priority
                                ;
                                ;  Finished
                                ;
001104  012602                  9$:     MOV     (SP)+,R2
001106  000207                          RETURN
```

```
                                ;*****************************************************************
                                ;  See if we need to suspend or resume a job.
                                ;
001110  010246                  CPSFLG: MOV     R2,-(SP)
                                ;
                                ;  See if any flags were specified
                                ;
001112  032767  000003  176700          BIT     #<CPFSUS!CPFRES>,CPFLAG ;Suspend or resume requested?
001120  001423                          BEQ     9$              ;Br if not
                                ;
                                ;  Make sure we aren't suspending or resuming our own job
                                ;
001122  116702  176675                  MOVB    CPPID,R2        ;Get ID # of job we want to affect
001126  120267  000000G                 CMPB    R2,CORUSR       ;Is it our job?
001132  001420                          BEQ     10$             ;Br if yes
                                ;
                                ;  Make sure we can access the specified job
                                ;
001134  004767  000000G                 CALL    CKACOJ          ;Can we access that job?
001140  103413                          BCS     9$              ;Br if not
                                ;
                                ;  Suspend or resume the specified job
                                ;
001142  012700  000000G                 MOV     #RJEMT,R0       ;Point to resume-job arg block
001146  032767  000002  176644          BIT     #CPFRES,CPFLAG  ;Does he want to resume the job?
001154  001002                          BNE     1$              ;Br if yes
001156  012700  000000G                 MOV     #SJEMT,R0       ;Point to suspend-job arg block
001162  010260  000004          1$:     MOV     R2,4(R0)        ;Set job # in EMT arg block
001166  104375                          EMT     375             ;Suspend or resume the job
                                ;
                                ;  Finished
                                ;
001170  012602                  9$:     MOV     (SP)+,R2
001172  000207                          RETURN
                                ;
                                ;  Error -- Trying to suspend or resume our own job
                                ;
001174                          10$:    FABORT  #EM$IDR         ;/ID parameter required
```

```
                                        .SBTTL  .       Terminal
                                ;-----------------------------------------------------------------
                                ;  Process the SET TERMINAL command.
                                ;
001204                          SETTTY:
                                ;
                                ;  See if a line number was specified
                                ;
001204  105067  176612                  CLRB    SETPRM          ;Assume no line number specified
001210  004767  0000000                 CALL    SKPSPC          ;Skip over any spaces
001214  010305                          MOV     R3,R5           ;Save pointer to first operand
001216  005002                          CLR     R2              ;Say no digits seen yet
001220  112500                  1$:     MOVB    (R5)+,R0        ;Get next char
001222  001413                          BEQ     2$              ;Br if hit end of command line
001224  120027  000040                  CMPB    R0,#40          ;Is this a space?
001230  001410                          BEQ     2$              ;Br if yes
001232  120027  000060                  CMPB    R0,#'0          ;Is this a digit?
001236  103435                          BLO     3$              ;Br if not
001240  120027  000071                  CMPB    R0,#'9
001244  101032                          BHI     3$              ;Br if not digit
001246  005202                          INC     R2              ;Remember we saw a digit
001250  000763                          BR      1$              ;Scan first item and see if it is a number
001252  005702                  2$:     TST     R2              ;Was first item a number?
001254  001426                          BEQ     3$              ;Br if not
                                ;
                                ;  There is a terminal number specified, accrue it
                                ;
001256  004767  0000000                 CALL    ACRDEC          ;Accrue the terminal number
                                ;
                                ;  See if this is a valid terminal number
                                ;
001262  006301                          ASL     R1              ;Convert line number to line index number
001264  001403                          BEQ     4$              ;Zero is not valid line number
001266  020127  0000000                 CMP     R1,#LSTPL       ;Is this a primary line number?
001272  101404                          BLOS    5$              ;Br if yes
001274                          4$:     FABORT  #EM$ILN         ;Invalid line number
                                ;
                                ;  Require TERMINAL privilege to make any permanent change
                                ;
001304  004767  0000000         5$:     CALL    CKTERM          ;Require TERMINAL privilege to do this
001310  110167  176506                  MOVB    R1,SETPRM       ;Remember to make set permanent
                                ;
                                ;  If we are changing line parameters for a line that is not
                                ;  currently logged on, initialize the LSW2 table from the initial
                                ;  flags for the line (ILSW2).
                                ;
001314  032761  0000000 0000000         BIT     #$DILUP,LSW(R1) ;Is line logged on?
001322  001003                          BNE     3$              ;Br if yes
001324  016161  0000000 0000000         MOV     ILSW2(R1),LSW2(R1);Initialize LSW2 table
                                ;
                                ;  We have the terminal line index number for which the command is being
                                ;  issued in R1.
                                ;  Process all qualifier specified with command.
                                ;
001332  012704  000000'         3$:     MOV     #TTHD,R4        ;Get pointer to option table
001336  004767  0000000                 CALL    SCNOPS          ;Process all of the command options
                                ;
```

```
                                        ;  End of command reached
                                        ;
001342  000167  0000006                         JMP     RDCMD           ;Finished SET TT command
```

```
                                ;*****************************************************************************
                                ;  SET TERMINAL n START
                                ;
001346  004767  000000G         STTTST: CALL    CKPRIV          ;Require OPER privilege
                                ;
                                ;  See if line is already started
                                ;
001352  032761  000000G 000000G        BIT     #$DILUP,LSW(R1) ;Is line already started?
001360  001404                         BEQ     1$              ;Br if not
001362                                 FABORT  #EM$LAS         ;Line is already started
                                ;
                                ;  See if line is marked dead
                                ;
001372  032761  000000G 000000G 1$:    BIT     #$DEAD,LSW3(R1) ;Is line marked as dead?
001400  001404                         BEQ     2$              ;Br if not
001402                                 FABORT  #EM$LFD         ;Line is dead
                                ;
                                ;  See if a CL unit is attached to the line
                                ;
001412  005761  000000G         2$:    TST     LCLUNT(R1)      ;Is line being used by a CL unit?
001416  002404                         BLT     3$              ;Br if not
001420                                 FABORT  #EM$CLU         ;Line in use by a CL unit
                                ;
                                ;  Start the line
                                ;
001430  010167  176474          3$:    MOV     R1,STAEMT+4     ;Store line index into EMT arg block
001434  012700  000124'                MOV     #STAEMT,R0      ;Point to EMT arg block
001440  104375                         EMT     375             ;Start the line
001442  000207                 9$:    RETURN
```

```
                                        ;-------------------------------------------------------------------------
                                        ;  SET TERMINAL [n] SPEED=n
                                        ;
001444  004767  000000G         STTTSP: CALL    CKTERM          ;Require TERMINAL privilege
001450  004767  000000G                 CALL    ACRSPD          ;Accrue speed value
001454  042761  000000G 000000G         BIC     #$AUTO,LSW2(R1) ;Say no autobaud on this line
001462  116100  000001G                 MOVB    LMXPRM+1(R1),R0 ;Get control flags for line
001466  042700  000000G                 BIC     #<LP$SPD>,R0    ;Clear speed flags
001472  050005                         BIS     R0,R5           ;Combine parity and length flags with speed
001474  010567  176372         STSP1:  MOV     R5,LSPEMT+4     ;Store speed, par, len code into EMT arg block
001500  010167  176364                 MOV     R1,LSPEMT+2     ;Store line # into EMT arg block
001504  006267  176360                 ASR     LSPEMT+2        ;Convert index # to line #
001510  012700  000066'                MOV     #LSPEMT,R0      ;Point to EMT arg block
001514  104375                         EMT     375             ;Set the speed
001516  103402                         BCS     2$              ;Br if error
001520  000167  000474                 JMP     BS01
                                        ;
                                        ;  Error on EMT to set speed
                                        ;
001524                          2$:     FABORT  #EM$ILN         ;Say invalid line number
```

```
                                ;-----------------------------------------------------------------
                                ;  SET TT [n] XON
                                ;
001534  004767  0000006         SETXON: CALL    CKTERM          ;Require TERMINAL privilege
001540  010167  176332                  MOV     R1,XONEMT+2     ;Set line # in EMT argument block
001544  006267  176326                  ASR     XONEMT+2        ;Convert index # to line #
001550  012700  000074'                 MOV     #XONEMT,R0      ;Point to EMT argument block
001554  104375                          EMT     375             ;Do the XON EMT
001556  103402                          BCS     2$              ;Br if invalid line #
001560  000167  000434                  JMP     BSO1
001564                          2$:     FABORT  #EM$ILN         ;Say invalid line number
```

```
                                        ;-----------------------------------------------------------------------
                                        ;  SET TT [n] [NO]DTR
                                        ;
001574  112767  000002  176276  SETDTR: MOVB    #2,DTREMT       ;Set function to raise DTR
001602  000403                        BR      DTRCOM          ;Br to common routine
001604  112767  000003  176266  CLRDTR: MOVB    #3,DTREMT       ;Set function to lower DTR
001612  004767  000000G         DTRCOM: CALL    CKTERM          ;Require TERMINAL privilege
001616  006201                        ASR     R1              ;Convert line index to line number
001620  010167  176256                  MOV     R1,DTREMT+2     ;Set in EMT  arg block
001624  006301                        ASL     R1              ;Get back as line index
001626  012700  000100'                MOV     #DTREMT,R0      ;Point to EMT arg block to
001632  104375                        EMT     375             ;Raise or lower DTR
001634  103402                        BCS     2$              ;Br on EMT error
001636  000167  000356                JMP     BSD1
001642                          2$:     FABORT  EM$ILN          ;Say invalid line number
```

```
                                ;-----------------------------------------------------------------
                                ;  SET TERMINAL AUTOBAUD
                                ;
001652  004767  000000G        SAUTO:  CALL    CKTERM              ;Require TERMINAL privilege
001656  052761  000000G 000000G        BIS     #$AUTO,LSW2(R1)     ;Set flag saying to do autobaude select
001664  032761  000000G 000000G        BIT     #$DILUP,LSW(R1)     ;Is anyone using the line now?
001672  001003                        BNE     9$                  ;Br if yes
001674  012705  000000G               MOV     #S9600,R5           ;Set line speed to 9600 baud
001700  000675                        BR      STSP1
001702  000167  000312        9$:     JMP     BSO1
```

```
                                ;-----------------------------------------------------------------------
                                ;  SET TERMINAL PARITY=(EVEN ODD NONE)
                                ;
001706  010446                  SETPAR: MOV     R4,-(SP)        ;Save current option table pointer
001710  004767  000000G                 CALL    CKTERM          ;Require TERMINAL privilege
001714  004767  000000G                 CALL    SKPSPC          ;Skip past any spaces
001720  121327  000075                  CMPB    (R3),#'=        ;Equal sign following PARITY?
001724  001001                          BNE     1$              ;Br if not
001726  005203                          INC     R3              ;Skip past equal sign
001730  012704  001134'          1$:     MOV     #PARHD,R4       ;Point to new option table
001734  004767  000000G                 CALL    SETWRD          ;Process EVEN, ODD, NONE word
001740  012604                          MOV     (SP)+,R4
001742  000207                          RETURN
```

```
                                        ;-------------------------------------------------------------------------
                                        ;  SET TERMINAL BITS=n
                                        ;
001744  004767  000000G        STCLEN: CALL    CKTERM              ;Require TERMINAL privilege
001750  116105  000001G                MOVB    LMXPRM+1(R1),R5     ;Get current line parameters
001754  010102                        MOV     R1,R2               ;Save line index number
001756  004767  000000G                CALL    ACRDEC              ;Accrue parameter value
001762  042705  000000G                BIC     #LP$7BT,R5          ;Assume 8 bits wanted
001766  020127  000007                CMP     R1,#7.              ;Is value 7?
001772  001003                        BNE     1$                  ;Br if not
001774  052705  000000G                BIS     #LP$7BT,R5          ;Select 7 bits
002000  000407                        BR      2$
002002  120127  000010        1$:     CMPB    R1,#8.              ;Is value 8?
002006  001404                        BEQ     2$                  ;Br if yes
002010                                FABORT  #EM$ICL             ;Invalid character length
002020  010201                2$:     MOV     R2,R1               ;Get back line index number
002022  000167  177446                JMP     STSP1               ;Go set new length
```

```
                                ;-----------------------------------------------------------------
                                ;  SET TERMINAL PARITY=(EVEN ODD NONE)
                                ;
                                ;  Inputs:
                                ;   R1 = line index number.
                                ;   R4 = Pointer to work with parity flag bits.
                                ;
002026  004767  000000G         STPAR:  CALL    CKTERM          ;Require TERMINAL privilege
002032  116105  000001G                 MOVB    LMXPRM+1(R1),R5 ;Get current flags for line
002036  042705  000000C                 BIC     #LP$PAR!LP$ODD,R5;Clear parity control flags
002042  151405                         BISB    @R4,R5          ;Set new flags
002044  000167  177424                 JMP     STSP1           ;Go set value
                                ;
                                ;  SET TERMINAL NOPARITY
                                ;
002050  004767  000000G         STNOPR: CALL    CKTERM          ;Require TERMINAL privilege
002054  116105  000001G                 MOVB    LMXPRM+1(R1),R5 ;Get current flags for line
002060  042705  000000C                 BIC     #LP$PAR!LP$ODD,R5;Clear parity control flags
002064  000167  177404                 JMP     STSP1           ;Go set value

                                ;-----------------------------------------------------------------
                                ;  SET TERMINAL [n] TRANSLATE=(ext=int,ext=int,...)
                                ;
002070  010146                 STTRNS: MOV     R1,-(SP)        ;Save original line number
002072  016101  000000G                 MOV     LNPRIM(R1),R1   ;Make sure we have primary line number
002076  004767  002344                 CALL    PRSTRN          ;Parse the translate qualifier
002102  012601                         MOV     (SP)+,R1        ;Restore original line number
002104  000207                         RETURN
```

```
                                ;-----------------------------------------------------------
                                ;  Set parameters for particular terminal types.
                                ;
                                ;  SET PARAMETERS FOR A VT50/VT52.
                                ;  SET VT50
002106  042761  000000G 000000G SVT50:   BIC     #VT52NO,LSW2(R1);CLEAR SOME FLAGS
002114  052761  000000G 000000G          BIS     #VT52FL,LSW2(R1);SAY TERMINAL IS A VT50
002122  012761  000000G 000000G          MOV     #VT52,LTRMTP(R1);SET TERMINAL TYPE
002130  000414                          BR      BSO2
                                ;
                                ;  SET PARAMETERS FOR LA36.
                                ;
002132  042761  000000G 000000G STLA36: BIC     #LA36NO,LSW2(R1);RESET VARIOUS FLAGS
002140  052761  000000G 000000G          BIS     #LA36FL,LSW2(R1)
002146  012761  000000G 000000G          MOV     #LA36,LTRMTP(R1);SET TERMINAL TYPE
002154  042761  000000G 000000G          BIC     #$SLON,LSW7(R1) ;Disalble SL editor for this terminal type
                                ;
                                ;  Set up terminal type information
                                ;
002162  120167  000000G          BSO2:   CMPB    R1,CORUSR       ;Changing terminal type for our job?
002166  001006                          BNE     1$              ;Br if not our job
002170  005761  000000G                 TST     LWINDO(R1)      ;Is process windowing turned on for job?
002174  001403                          BEQ     1$              ;Br if not
002176  012700  000132'                 MOV     #WINSTT,R0      ;Point to EMT argument block
002202  104375                          EMT     375             ;Tell windowing system about terminal type
002204  105767  175612          1$:     TSTB    SETPRM          ;Should we make the set permanent?
002210  001403                          BEQ     BSO1            ;Br if not
002212  016161  000000G 000000G          MOV     LTRMTP(R1),ITRMTP(R1) ;Set initial terminal type
002220  016161  000000G 000000G BSO1:    MOV     LSW2(R1),LSW2S(R1) ;Save in case of ctrl-c reentry
002226  105767  175570                  TSTB    SETPRM          ;Should we make the set permanent?
002232  001403                          BEQ     SETJMP          ;Br if not
002234  016161  000000G 000000G          MOV     LSW2(R1),ILSW2(R1);Set initial flags for line
002242  000207                  SETJMP: RETURN                  ;Return to process next qualifier
                                ;
                                ;  LA120 parameters
                                ;
002244  042761  000000G 000000G SLA120: BIC     #LA12NO,LSW2(R1);SET FLAGS
002252  052761  000000G 000000G          BIS     #LA12FL,LSW2(R1)
002260  012761  000000G 000000G          MOV     #LA120,LTRMTP(R1);SET TERMINAL TYPE
002266  042761  000000G 000000G          BIC     #$SLON,LSW7(R1) ;Disalble SL editor for this terminal type
002274  000732                          BR      BSO2
                                ;
                                ;  VT100 parameters
                                ;
002276  042761  000000G 000000G SVT100: BIC     #VT10NO,LSW2(R1);SET FLAGS
002304  052761  000000G 000000G          BIS     #VT10FL,LSW2(R1)
002312  012761  000000G 000000G          MOV     #VT100,LTRMTP(R1);SET TERMINAL TYPE
002320  000720                  STERM:  BR      BSO2
                                ;
                                ;  VT200 parameters
                                ;
002322  042761  000000G 000000G SVT200: BIC     #VT20NO,LSW2(R1)
002330  052761  000000G 000000G          BIS     #VT20FL,LSW2(R1)
002336  012761  000000G 000000G          MOV     #VT200,LTRMTP(R1)
002344  000706                          BR      BSO2
                                ;
                                ;  VT200 parameters 7 bit controls
```

```
                                        ;
002346  042761  000000G 000000G SVT227: BIC     #VT20NO,LSW2(R1)
002354  052761  000000G 000000G         BIS     #VT20FL,LSW2(R1)
002362  012761  000000G 000000G         MOV     #VT2007,LTRMTP(R1)
002370  000674                         BR      BS02
                                        ;
                                        ;  VT200 parameters 8 bit controls
                                        ;
002372  042761  000000G 000000G SVT228: BIC     #VT20NO,LSW2(R1)
002400  052761  000000G 000000G         BIS     #VT20FL,LSW2(R1)
002406  012761  000000G 000000G         MOV     #VT2008,LTRMTP(R1)
002414  000662                         BR      BS02
                                        ;
                                        ;  Hazeltine parameters
                                        ;
002416  042761  000000G 000000G STHAZL: BIC     #HAZLNO,LSW2(R1);RESET SOME FLAGS
002424  052761  000000G 000000G         BIS     #HAZLFL,LSW2(R1);SET SOME FLAGS
002432  012761  000000G 000000G         MOV     #HAZEL,LTRMTP(R1);SET TERMINAL TYPE
002440  042761  000000G 000000G         BIC     #$SLON,LSW7(R1) ;Disalble SL editor for this terminal type
002446  000645                         BR      BS02
                                        ;
                                        ;  ADM3A parameters
                                        ;
002450  042761  000000G 000000G SADM3A: BIC     #ADM3NO,LSW2(R1);SET FLAGS
002456  052761  000000G 000000G         BIS     #ADM3FL,LSW2(R1)
002464  012761  000000G 000000G         MOV     #ADM3A,LTRMTP(R1);SET TERMINAL TYPE
002472  042761  000000G 000000G         BIC     #$SLON,LSW7(R1) ;Disalble SL editor for this terminal type
002500  000630                         BR      BS02
                                        ;
                                        ;  SET PARAMETERS FOR DIABLO-1620 TERMINAL
                                        ;
002502                          SQUME:
002502  042761  000000G 000000G SDIAB:  BIC     #DIABNO,LSW2(R1);RESET VARIOUS FLAGS
002510  052761  000000G 000000G         BIS     #DIABFL,LSW2(R1);SAY THIS IS A DIABLO
002516  012761  000000G 000000G         MOV     #DIABLO,LTRMTP(R1);SET TERMINAL TYPE
002524  042761  000000G 000000G         BIC     #$SLON,LSW7(R1) ;Disalble SL editor for this terminal type
002532  000613                         BR      BS02
                                        ;
                                        ;  SET TT QUIET
                                        ;  (Also, SET NOVERIFY)
                                        ;
002534  116701  000000G         STNOVR: MOVB    CORUSR,R1               ;Get current job index number
002540  004767  000004                 CALL    SETQUT                  ;Do the set
002544  000167  000000G                JMP     RDCMD                   ;Finished
                                        ;
002550  052761  000000G 000000G SETQUT: BIS     #$QTSET,LSW2(R1)        ;REMEMBER SET WAS DONE
002556  052761  000000G 000000G         BIS     #$QUIET,LSW4(R1)        ;HAVE IMMEDIATE EFFECT
002564  120167  000000G                CMPB    R1,CORUSR               ;Are we doing set for current job?
002570  001025                         BNE     1$                      ;Br if not
002572                                 .GVAL   #XAREA,#<CFSTS-MONVEC>  ;GET CURRENT COMMAND FILE FLAGS
002612  010002                         MOV     R0,R2
002614  052702  000000G                BIS     #CF$QUT,R2              ;SET TT-QUIET FLAG
002620                                 .PVAL   #XAREA,#<CFSTS-MONVEC>,R2;STORE UPDATED FLAGS
002644  000167  177350         1$:     JMP     BS01
                                        ;
                                        ;  SET TT NOQUIET
                                        ;  (Also, SET VERIFY)
```

```
                                ;
002650  116701  000000G          STVRFY: MOVB    CORUSR,R1               ;Get current job index number
002654  004767  000004                  CALL    RSTQUT                  ;Do the SET
002660  000167  000000G                  JMP     RDCMD                   ;Finished
                                ;
002664  042761  000000G 000000G RSTQUT: BIC     #$QTSET,LSW2(R1)        ;REMEMBER SET WAS DONE
002672  042761  000000G 000000G         BIC     #$QUIET,LSW4(R1)        ;START LISTING NOW
002700  120167  000000G                  CMPB    R1,CORUSR               ;Are we doing set for current line?
002704  001025                          BNE     1$                      ;Br if not
002706                                  .GVAL   #XAREA,#<CFSTS-MONVEC>  ;GET CURRENT COMMAND FILE FLAGS
002726  010002                          MOV     R0,R2
002730  042702  000000G                BIC     #CF$QUT,R2              ;CLEAR TT-QUIET FLAG
002734                                  .PVAL   #XAREA,#<CFSTS-MONVEC>,R2;STORE UPDATED FLAGS
002760  000167  177234          1$:     JMP     BSO1
                                ;
                                ;   SET TT NOSCOPE
                                ;
002764  042761  000000G 000000G SETNSC: BIC     #$SCOPE,LSW2(R1)        ;Say this is not a scope type terminal
002772  042761  000000G 000000G         BIC     #$SLON,LSW7(R1)         ;Disable SL editor
003000  000167  177214                  JMP     BSO1
                                ;
                                ;   SET TT WAIT
                                ;
003004  042761  000000G 000000G SETTW:  BIC     #$SNWTT,LSW6(R1);UNDO SET TT NOWAIT
003012  042761  000000G 000000G         BIC     #$NOWTT,LSW5(R1)
003020  000207                          RETURN
                                ;
                                ;   SET TT NOWAIT
                                ;
003022  052761  000000G 000000G SETTNW: BIS     #$SNWTT,LSW6(R1);SET TT NOWAIT
003030  052761  000000G 000000G         BIS     #$NOWTT,LSW5(R1)
003036  000207                          RETURN
                                ;
                                ;   SET TT DEAD
                                ;
003040  004767  000000G          SETDED: CALL    CKTERM                  ;Require TERMINAL privilege
003044  032761  000000G 000000G         BIT     #$DILUP,LSW(R1)         ;Is line active now
003052  001406                          BEQ     SETTTB                  ;Br if not
003054                                  FABORT  #RUNMS                  ;Can't kill an active line
                                ;
                                ;   Perform terminal sets that only involve setting or clearing flag bits.
                                ;
003064  004767  000000G          SETTTP: CALL    CKTERM                  ;Require TERMINAL privilege
003070  012405                  SETTTB: MOV     (R4)+,R5                ;POINT TO TABLE
003072  060105                          ADD     R1,R5                   ;POINT TO TABLE ENTRY FOR THIS USER
003074  051415                          BIS     @R4,@R5                 ;SET THE DESIRED FLAG
003076  000167  177116                  JMP     BSO1
                                ;
                                ;   RESET A BIT IN A USER TABLE
                                ;
003102  004767  000000G          CLRTTP: CALL    CKTERM                  ;Require TERMINAL privilege
003106  012405                  CLRTTB: MOV     (R4)+,R5                ;POINT TO TABLE
003110  060105                          ADD     R1,R5                   ;POINT TO TABLE ENTRY FOR THIS USER
003112  041415                          BIC     @R4,@R5                 ;RESET THE DESIRED FLAG
003114  000167  177100                  JMP     BSO1
```

```
                                        .SBTTL  .       CL
                                ;-----------------------------------------------------------------------
                                ;  Process the SET CLn command
                                ;
                                ;  Inputs:
                                ;   R2 = CL unit number
                                ;
003120  004767  000000G         SETCL:  CALL    CKTERM          ;Require TERMINAL privilege for SET CL
003124  020227  000000G                 CMP     R2,#CLTOTL      ;Is this a valid unit?
003130  103404                         BLO     3$              ;Br if ok
003132                                  FABORT  #EM$IUN         ;Invalid CL unit
003142  006302                 3$:     ASL     R2              ;Convert unit # to word table index
                                ;
                                ;  At this point, R2 contains the index into the CL unit tables.
                                ;  Begin loop to process each specified command qualifier.
                                ;
003144  004767  000000G         1$:     CALL    SKPSPC          ;Skip over any spaces
003150  111300                         MOVB    (R3),R0         ;Get next character from command
003152  001443                         BEQ     15$             ;Br if hit the end of the command
003154  120027  000054                 CMPB    R0,#',          ;Comma separator?
003160  001403                         BEQ     4$              ;Br if yes
003162  120027  000057                 CMPB    R0,#'/          ;Slash separator?
003166  001005                         BNE     5$              ;Br if not
003170  005203                 4$:     INC     R3              ;Skip past separator
003172  004767  000000G                 CALL    SKPSPC          ;Skip over any spaces
003176  105713                         TSTB    (R3)            ;Are we at the end of the command now?
003200  001424                         BEQ     10$             ;Br if yes -- Error
                                ;
                                ;  Process the next command qualifier
                                ;
003202  012704  001170'         5$:     MOV     #CLOPHD,R4      ;Point to option table
003206  004767  000000G                 CALL    SEARCH          ;Search for option keyword
003212  103413                         BCS     11$             ;Br if cannot idenify keyword
003214  020427  001624'                 CMP     R4,#CLNCKA      ;Does this set command need to check alloc?
003220  101004                         BHI     51$             ;BR if not
003222  016700  000000G                 MOV     R50BUF,R0       ;Get back RAD50 device name
003226  004767  000000G                 CALL    CHKALC          ;Forbid SET if device is allocated
003232  010246                 51$:    MOV     R2,-(SP)        ;Save unit index
003234  004734                         CALL    @(R4)+          ;Call routine to process qualifier
003236  012602                         MOV     (SP)+,R2        ;Get back unit number index
003240  000741                         BR      1$              ;Go see if there are more qualifiers
                                ;
                                ;  Invalid qualifier keyword
                                ;
003242                          11$:    FABORT  #CSIMS1         ;Invalid option keyword
                                ;
                                ;  Invalid command syntax
                                ;
003252                          10$:    FABORT  #EM$CSE         ;Command syntax error
                                ;
                                ;  End of command reached
                                ;
003262  000167  000000G         15$:    JMP     RDCMD           ;Finished command
                                ;
                                ;  CL unit is not assigned to a line
                                ;
003266                          STCLNL: FABORT  #EM$CLN         ;CL unit is not assigned to a line
```

```
                                ;
                                ;  Turn on some option
                                ;
003276  005762  000000G        STCLOP: TST     CL$LIX(R2)       ;Is this CL unit assigned to a line?
003302  001771                        BEQ     STCLNL           ;Br if not
003304  051462  000000G                BIS     (R4),CL$OPT(R2) ;Set desired option flag
003310  000207                        RETURN
                                ;
                                ;  Turn off some option
                                ;
003312  005762  000000G        CLCLOP: TST     CL$LIX(R2)       ;Is this CL unit assigned to a line?
003316  001763                        BEQ     STCLNL           ;Br if not
003320  041462  000000G                BIC     (R4),CL$OPT(R2) ;Reset the option
003324  000207                        RETURN
                                ;
                                ;  Set a parameter value
                                ;
003326  005762  000000G        STCLVL: TST     CL$LIX(R2)       ;Is this CL unit assigned to a line?
003332  001755                        BEQ     STCLNL           ;Br if not
003334  004767  000000G                CALL    ACRDEC           ;Accrue decimal value
003340  011404                        MOV     (R4),R4          ;Get address of table to store value into
003342  060204                        ADD     R2,R4            ;Point to correct table entry
003344  010114                        MOV     R1,(R4)          ;Store value into table
003346  000207                        RETURN
                                ;
                                ;  SET CL [NO]GRAPH
                                ;
003350  005762  000000G        STCLGR: TST     CL$LIX(R2)       ;Is this CL unit assigned to a line?
003354  001744                        BEQ     STCLNL           ;Br if not
003356  011462  000000G                MOV     (R4),CL$WID(R2) ;Set width to 0 (graph) or 132 (nograph)
003362  000207                        RETURN
                                ;
                                ;  SET CL TOP
                                ;
003364  005762  000000G        CLTOP:  TST     CL$LIX(R2)       ;Is this CL unit assigned to a line?
003370  001736                        BEQ     STCLNL           ;Br if not
003372  005062  000000G                CLR     CL$LIN(R2)       ;Say we are at top of the page
003376  000207                        RETURN
                                ;
                                ;  SET CL SPEED=value
                                ;
003400  016202  000000G        STCLSP: MOV     CL$LIX(R2),R2    ;Is this CL unit assigned to a line?
003404  001730                        BEQ     STCLNL           ;Br if not
003406  004767  000000G                CALL    ACRSPD           ;Accrue speed value
003412  116200  000001G                MOVB    LMXPRM+1(R2),R0 ;Get current flags for line
003416  042700  000000G                BIC     #LP$SPD,R0       ;Clear speed code
003422  050005                        BIS     R0,R5            ;Put in parity and length flags
003424  010567  174442        STCLS1: MOV     R5,LSPEMT+4      ;Store speed code into EMT arg block
003430  006202                        ASR     R2               ;Convert line index # to line #
003432  010267  174432                MOV     R2,LSPEMT+2      ;Store line # into EMT arg block
003436  012700  000066'               MOV     #LSPEMT,R0       ;Point to EMT arg block
003442  104375                        EMT     375              ;Set the speed
003444  103401                        BCS     2$               ;Br if error
003446  000207                        RETURN
                                ;
                                ;  Error on EMT to set speed
                                ;
```

```
    115 003450                          2$:     FABORT  #EM$IUN         ;Say invalid unit number
    116                                 ;
    117                                 ;  SET CLn XON
    118                                 ;
    119 003460  012705  000000G         STCLXN: MOV     #CLSFCH,R5      ;Get .SPFUN code
    120 003464  000402                          BR      CLSPFN          ;Do the .SPFUN
    121                                 ;
    122                                 ;  SET CLn RESET
    123                                 ;
    124 003466  012705  000000G         STCLRS: MOV     #CLSFRS,R5      ;Get .SPFUN code
    125                                 ;
    126                                 ;  Do a .SPFUN to a CL unit.
    127                                 ;  R5 = .SPFUN code.
    128                                 ;
    129 003472  005762  000000G         CLSPFN: TST     CL$LIX(R2)      ;Is this CL unit assigned to a line?
    130 003476  001673                          BEQ     STCLNL          ;Br if not
    131 003500  004767  001140                  CALL    CLLOOK          ;Open channel 1 to CL unit
    132 003504                                  .SPFUN  #XAREA,#1,R5,#0,#0,#0 ;Do the function
    133 003550  103404                          BCS     2$              ;Br if error on .SPFUN
    134 003552                                  .CLOSE  #1              ;Close CL channel
    135 003560  000207                          RETURN
    136 003562                          2$:     .PURGE  #1              ;Make sure channel is closed
    137 003570                                  FABORT  #EM$IUN         ;Say invalid unit number
    138                                 ;
    139                                 ;  SET CL PARITY  driver routine
    140                                 ;
    141 003600  004767  000000G         STCLPO: CALL    SKPSPC          ;Skip any spaces
    142 003604  122327  000075                  CMPB    (R3)+,#'=       ;Should have an equal sign
    143 003610  001401                          BEQ     1$
    144 003612  005303                          DEC     R3              ;Point to start of keyword
    145 003614  012704  001642'         1$:     MOV     #CLPRHD,R4      ;Point to driver table
    146 003620  004767  000000G                 CALL    SETWRD          ;Process the option
    147 003624  000207                          RETURN
```

```
                                ;-------------------------------------------------------------------------
                                ;  BYPASS forms of SET CLn XON and RESET commands.
                                ;  These forms differ from the ordinary forms in that they use
                                ;  EMT processing instead of SPFUN calls to the CL handler, and thus
                                ;  do not have the overhead or access checking of getting a channel.
                                ;
                                ;  SET CLn XONBYPASS    Same as SET CLn XON, except uses EMT instead of SPFUN
                                ;
                                ;  R2 contains CL unit index
                                ;
003626  004767  000040          STCLXB: CALL    CHKBCM          ;Make common checks for SET CLn xxxBYPASS cmds
003632  006202                          ASR     R2              ;Convert CL unit index back into unit number
003634  012700  000050'                 MOV     #CLXEMT,R0      ;Point to emt arg block to set cln xon
003640  010260  000002                  MOV     R2,2(R0)        ;Set CL unit number into arg block
003644  104375                          EMT     375             ;Issue the command
003646  000207                          RETURN
                                ;
                                ;  SET CLn RESETBYPASS  Same as SET CLn RESETBYPASS, except uses EMT method
                                ;
                                ;  R2 contains CL unit index
                                ;
003650  004767  000016          STCLRB: CALL    CHKBCM          ;Make common checks for SET CLn xxxBYPASS cmds
003654  006202                          ASR     R2              ;Convert CL unit index back into unit number
003656  012700  000054'                 MOV     #CLREMT,R0      ;Point to EMT arg block to SET CLn RESET
003662  010260  000002                  MOV     R2,2(R0)        ;Set CL unit number into arg block
003666  104375                          EMT     375
003670  000207                          RETURN
                                ;
                                ;  Common check routine for SET CLn XONBYPASS and RESETBYPASS
                                ;  Inputs:
                                ;       R2      contains CL unit index
                                ;
003672  032767  000000G 000000G CHKBCM: BIT     #PO$BYP,PRIVCO  ;Are we privileged to use this command?
003700  001004                          BNE     1$              ;BR if OK
003702                                  FABORT  #EM$BYP         ;You must have BYPASS privilege for this cmd
003712  005762  000000G           1$:   TST     CL$LIX(R2)      ;Is specified CL unit connected to a line?
003716  001002                          BNE     2$              ;Br if so
003720  000167  177342                  JMP     STCLNL          ;Else abort if not connected to a line
003724  000207                  2$:     RETURN
```

```
                                ;-----------------------------------------------------------------------------
                                ;  SET CL PARITY=(EVEN ODD NONE)
                                ;
                                ;  Inputs:
                                ;   R2 = CL unit number index
                                ;   R4 = Pointer to word with parity control flags
                                ;
003726  011404                  STCLPR: MOV     (R4),R4          ;Get parity flags
003730  016202  0000006         STCLP1: MOV     CL$LIX(R2),R2    ;Get number of line CL unit is connected to
003734  001406                          BEQ     9$               ;Br if not assigned to a line
003736  116205  0000016                 MOVB    LMXPRM+1(R2),R5 ;Get current control flags for line
003742  042705  000000C                 BIC     #LP$PAR!LP$ODD,R5 ;Clear parity control flags
003746  050405                          BIS     R4,R5            ;Put in new flags
003750  000625                          BR      STCLS1           ;Go set value for line
003752                          9$:     FABORT  #EM$CLN          ;CL unit is not assigned to a line
                                ;
                                ;  SET CL NOPARITY
                                ;
003762  005004                  STCLNP: CLR     R4               ;Clear parity flags
003764  000761                          BR      STCLP1           ;Go do the set
```

```
                                ;-------------------------------------------------------------------------------
                                ;  SET CL BITS=n
                                ;
003766  016202  0000006 STCLBT: MOV     CL$LIX(R2),R2   ;Get index for line CL unit is assigned to
003772  001425                 BEQ     9$              ;Br if not assigned to a line
003774  004767  0000006        CALL    ACRDEC          ;Accrue parameter value
004000  116205  0000016        MOVB    LMXPRM+1(R2),R5 ;Get current flags for line
004004  042705  0000006        BIC     #LP$7BT,R5      ;Assume 8 bits wanted
004010  020127  000007         CMP     R1,#7.          ;7 bits wanted?
004014  001003                 BNE     1$              ;Br if not
004016  052705  0000006        BIS     #LP$7BT,R5      ;Set 7 bit flag
004022  000407                 BR      2$
004024  020127  000010 1$:     CMP     R1,#8.          ;8 bits wanted?
004030  001404                 BEQ     2$              ;Br if yes
004032                         FABORT  #EM$ICL         ;Invalid character length
004042  000167  177356 2$:     JMP     STCLS1          ;Go set value
004046                 9$:     FABORT  #EM$CLN         ;CL unit not assigned to line
```

```
                        ;-------------------------------------------------------------------------------
                        ;  SET CL LINE=n
                        ;
004056                  STCLLN:
                        ;
                        ;  Accrue the line number
                        ;
004056  004767  000000G         CALL    ACRDEC          ;Accrue decimal value
                        ;
                        ;  Do the EMT to assign this CL unit to a line and check for errors.
                        ;
004062  006202                  ASR     R2              ;Convert CL unit index to unit #
004064  010267  173754          MOV     R2,CLAEMT+2     ;Set CL unit number
004070  010167  173752          MOV     R1,CLAEMT+4     ;Set time-sharing line number
004074  012700  000042'          MOV     #CLAEMT,R0      ;Point to EMT argument block
004100  104375                  EMT     375             ;Do the EMT
004102  103045                  BCC     20$             ;Br if no error
                        ;
                        ;  An error occurred on the EMT.
                        ;  Print an error message
                        ;
004104  113702  000000G         MOVB    @#ERRLOC,R2     ;Get EMT error code
004110  120227  000002          CMPB    R2,#2.          ;Invalid CL unit # ?
004114  001004                  BNE     3$              ;Br if not
004116                          FABORT  #EM$IUN         ;Invalid CL unit number
004126  120227  000003  3$:     CMPB    R2,#3.          ;Invalid line number?
004132  001004                  BNE     4$              ;Br if not
004134                          FABORT  #EM$ILN         ;Invalid line number
004144  120227  000004  4$:     CMPB    R2,#4.          ;Line already in use by a CL unit?
004150  001004                  BNE     5$              ;Br if not
004152                          FABORT  #EM$ACL         ;Line already being used by CL
004162  120227  000005  5$:     CMPB    R2,#5.          ;Is line in use by a time-sharing user?
004166  001004                  BNE     6$              ;Br if not
004170                          FABORT  #EM$TSL         ;Line busy as time-sharing line
004200  120227  000006  6$:     CMPB    R2,#6.          ;Is this CL unit currently busy?
004204  001004                  BNE     20$             ;Br if not
004206                          FABORT  #EM$CLB         ;This CL unit is busy
                        ;
                        ;  Finished
                        ;
004216  000207          20$:    RETURN
```

```
                                ;-----------------------------------------------------------------------------
                                ;  SET CL VERSION=n
                                ;
004220  004767  000000G        STCLVR: CALL    ACRDEC          ;Accrue parameter value
004224  120127  000017                CMPB    R1,#15.         ;Is this a valid argument
004230  002403                        BLT     9$              ;Br if not
004232  110167  000000G               MOVB    R1,CLVERS       ;Set version number
004236  000404                        BR      10$             ;Return
004240                          9$:     FABORT  #EM$IVN         ;Invalid version number
                                ;
                                ;  Finished
                                ;
004250  000207                  10$:     RETURN
```

```
                        ;-----------------------------------------------------------------------------
                        ;  SET CLn ENDSTRING='string'
                        ;
004252  005762  000000G CLESTR: TST     CL$LIX(R2)      ;Is this CL unit assigned to a line?
004256  001002                 BNE     2$              ;Br if yes
004260  000167  177002         JMP     STCLNL          ;Br if not
004264  010446          2$:    MOV     R4,-(SP)
004266  010546                 MOV     R5,-(SP)
                        ;
                        ;  Accrue the string
                        ;
004270  004767  000000G        CALL    ACRSTR          ;Accrue the string
                        ;
                        ;  Make sure the string is not too long
                        ;
004274  020027  000000G        CMP     R0,#CLEOFS      ;Is string too long?
004300  101046                 BHI     10$             ;Br if yes
                        ;
                        ;  Move string over in BLKO to allow room for ENDPAGE value in 1st word
                        ;
004302  012705  000000G        MOV     #BLKO,R5        ;Point to start of buffer
004306  005200                 INC     R0              ;Get length of string with null
004310  060005                 ADD     R0,R5           ;Point past last char in string
004312  010504                 MOV     R5,R4
004314  062704  000002         ADD     #2,R4           ;Add 1 word offset
004320  114544          3$:    MOVB    -(R5),-(R4)     ;Move string over
004322  077002                 SOB     R0,3$
004324  012767  000377  000000G MOV     #377,BLKO       ;Set value for ENDPAGE (377=no change)
                        ;
                        ;  Do lookup on CL unit
                        ;
004332  004767  000306         CALL    CLLOOK          ;Do lookup on channel 1 to CL unit
                        ;
                        ;  Perform the .SPFUN to set the ENDSTRING
                        ;
004336                         .SPFUN  #XAREA,#1,#CLSFEP,#BLKO,#10.,#0
004402                         .CLOSE  #1              ;Close the channel
                        ;
                        ;  Finished
                        ;
004410  012605                 MOV     (SP)+,R5
004412  012604                 MOV     (SP)+,R4
004414  000207                 RETURN
                        ;
                        ;  String is too long
                        ;
004416                 10$:    FABORT  #EM$STL         ;String too long
```

```
                                ;-----------------------------------------------------------------------------
                                ;  SET CLn TRANSLATE=(ext=int,ext=int,...)
                                ;
004426  016201  000000G       CLTRNS: MOV     CL$LIX(R2),R1   ;Is this CL unit assigned to a line?
004432  001002                  BNE     2$              ;Br if yes
004434  000167  176626          JMP     STCLNL          ;Br if not
004440  004767  000002        2$:     CALL    PRSTRN          ;Parse the translate qualifier
004444  000207                  RETURN
                                ;-----------------------------------------------------------------------------
                                ;  Parse a translate qualifier of the form:
                                ;       TRANSLATE=(ext=int,ext=int,...)
                                ;  and store the translation table for the line.
                                ;
                                ;  Inputs:
                                ;   R1 = Line index number of line whose translation table is being set up.
                                ;   R3 = Pointer past end of keyword
                                ;
004446  010446                  PRSTRN: MOV     R4,-(SP)
004450  010546                  MOV     R5,-(SP)
                                ;
                                ;  Get pointer to translation table area for this line
                                ;
004452  016104  000000G         MOV     LCXTBL(R1),R4   ;Get pointer to translation table for line
004456  001004                  BNE     7$              ;Br if there is one
004460                          FABORT  #EM$TMT         ;No translation table allocated
                                ;
                                ;  See if equal sign and open paren follow keyword
                                ;
004470  004767  000000G         7$:     CALL    SKPSPC          ;Skip over any spaces
004474  122327  000075          CMPB    (R3)+,#'=       ;Does equal sign follow keyword?
004500  001406                  BEQ     1$              ;Br if yes
004502  126327  177777  000050  CMPB    -1(R3),#'(      ;Did user omit equal sign?
004510  001413                  BEQ     2$              ;Br if yes
004512  005303                  DEC     R3              ;Point back to character
004514  000447                  BR      3$              ;Clear the translation table
004516  004767  000000G         1$:     CALL    SKPSPC          ;Skip more spaces
004522  122327  000050          CMPB    (R3)+,#'(       ;Should have open paren
004526  001404                  BEQ     2$              ;Br if got open paren
004530                          FABORT  #EM$CSE         ;Invalid syntax
                                ;
                                ;  Begin to accrue each value pair
                                ;
004540  005005                  2$:     CLR     R5              ;Count pairs in R5
                                ;
                                ;  See if we have reached the end of the list
                                ;
004542  004767  000000G         5$:     CALL    SKPSPC          ;Skip any spaces
004546  122327  000051          CMPB    (R3)+,#')       ;At end of the list?
004552  001430                  BEQ     3$              ;Br if yes
004554  005303                  DEC     R3              ;Point back to character
                                ;
                                ;  See if we have room for another value
                                ;
004556  020527  000000G         CMP     R5,#MXTTCT      ;Room for another value?
004562  103404                  BLO     4$              ;Br if yes
004564                          FABORT  #EM$TMT         ;Too many translation values
```

```
                                4$:     INC     R5              ;Count another value
                                ;
                                ;  Accrue the external value
                                ;
                                        CALL    ACROCT          ;Accrue the external value
                                        MOVB    R1,1(R4)        ;Store in high-order byte of word
                                ;
                                ;  Accrue the internal value
                                ;
                                        CALL    ACROCT          ;Accrue the internal value
                                        MOVB    R1,(R4)         ;Store in low-order byte of word
                                        TST     (R4)+           ;Point to next word
                                        CALL    SKPSPC          ;Skip any spaces
                                        CMPB    (R3)+,#',       ;Comma separator?
                                        BEQ     5$              ;Br if yes
                                        DEC     R3              ;Point back to char we skipped
                                        BR      5$              ;Loop and get rest of list
                                ;
                                ;  We have reached the end of the list
                                ;
                                3$:     CLR     (R4)            ;Store a zero to terminate the list
                                ;
                                ;  Finished
                                ;
                                        MOV     (SP)+,R5
                                        MOV     (SP)+,R4
                                        RETURN
```

```
                                ;-------------------------------------------------------------------------------
                                ;  Perform a lookup on channel 1 to a specified CL unit.
                                ;
                                ;  Inputs:
                                ;   R2 = CL unit index number.
                                ;
                                ;  Outputs:
                                ;   Channel 1 opened to CL unit.
                                ;
004644  010246                  CLLOOK: MOV     R2,-(SP)
                                ;
                                ;  Construct CL device name
                                ;
004646  006202                          ASR     R2              ;Get cl unit number
004650  020227  000007                  CMP     R2,#7.          ;Should this be a CL or C1 unit?
004654  101405                          BLOS    2$              ;Br if CL
004656  162702  000010                  SUB     #8.,R2          ;Remove C1 unit bias
004662  066702  173130                  ADD     R50C10,R2       ;Add "C1O" to form device name
004666  000402                          BR      1$
004670  066702  173120          2$:     ADD     R50CLO,R2       ;Add "CLO" to form device name
004674  010267  173104          1$:     MOV     R2,CLDEV        ;Save device name
004700                                  .LOOKUP #XAREA,#1,#CLDEV;Open channel to CL unit
004720  103402                          BCS     10$             ;Br if error on lookup
                                ;
                                ;  Finished
                                ;
004722  012602                          MOV     (SP)+,R2
004724  000207                          RETURN
                                ;
                                ;  Error on lookup to CL unit
                                ;
004726                          10$:    FABORT  #EM$IUN         ;Invalid unit number
```

```
                                        .SBTTL  .       Host
                                ;-------------------------------------------------------------------------
                                ;  SET HOST/DTE=n
                                ;  Cross connect time-sharint line with CL unit.
                                ;
004736  004767  000000G         SETHST: CALL    CKTERM          ;Require TERMINAL privilege for SET HOST
                                ;
                                ;  Do command scanning
                                ;
004742  012767  177777  173114          MOV     #-1,TTXCL+4     ;Initially set CL unit to -1
004750  012704  001670'                 MOV     #HOSTHD,R4      ;Point to option driver table
004754  004767  000000G                 CALL    SCNOPS          ;Process the command options
                                ;
                                ;  Check to make sure the CL unit is assigned to a line and is not
                                ;  in use by another user.
                                ;
004760  016705  173100                  MOV     TTXCL+4,R5      ;Get CL unit number
004764  020527  000000G                 CMP     R5,#CLTOTL      ;Is this a valid unit number
004770  103404                          BLO     1$              ;Br if yes
004772                                  FABORT  #EM$IUN         ;Invalid CL unit number
005002  006305                  1$:     ASL     R5              ;Convert to CL index number
005004  010567  173054                  MOV     R5,TTXCL+4
005010  005765  000000G                 TST     CL$LIX(R5)      ;Is this CL unit connected to a line?
005014  001004                          BNE     2$              ;Br if yes
005016                                  FABORT  #EM$CLN         ;Not connected to line
005026  010500                  2$:     MOV     R5,R0           ;Get CL unit index
005030  004767  000000G                 CALL    CKCLUS          ;See if CL unit in use by another job
005034  006300                          ASL     R0              ;Any job using CL unit?
005036  001407                          BEQ     3$              ;Br if not
005040  120067  000000G                 CMPB    R0,CORUSR       ;Is it our job?
005044  001404                          BEQ     3$              ;Br if yes
005046                                  FABORT  #EM$CLB         ;This CL unit is busy
                                ;
                                ;  Perform the connection
                                ;
005056  116701  000000G         3$:     MOVB    CORUSR,R1       ;Get our job index number
005062  016761  172712  000000G         MOV     R50HST,LPRG1(R1);Set name of running program to "$HOST$"
005070  016761  172706  000000G         MOV     R50HST+2,LPRG2(R1)
005076  012700  000060'                 MOV     #TTXCL,R0       ;Point to EMT argument block
005102  104375                          EMT     375             ;Make the cross connection
                                ;
                                ;  Print message saying cross connection broken
                                ;
005104  016761  000000G 000000G         MOV     R50KMN,LPRG1(R1);Reset program name to KMON
005112  016761  000002G 000000G         MOV     R50KMN+2,LPRG2(R1)
005120                                  .PRINT  #TM$XBK
005126  000167  000000G                 JMP     RDCMD           ;Finished with command
                                ;
                                ;  Process the /PORT=ddn qualifier
                                ;
005132  010246                  HSTPRT: MOV     R2,-(SP)
005134  004767  000000G                 CALL    CHKEQ           ;Equal sign should follow qualifier name
005140  004767  000000G                 CALL    GTRD50          ;Accrue the device name
005144  122327  000072                  CMPB    (R3)+,#':       ;Colon specified with device name?
005150  001401                          BEQ     1$              ;Br if yes
005152  005303                          DEC     R3              ;Backup pointer
005154  016700  000000G         1$:     MOV     R50BUF,R0       ;Get accrued device name
```

```
     58 005160  004767  0000006                    CALL    ASNSRC          ;See if this is a logical device name
     59 005164  103402                             BCS     2$              ;Br if not logical name
     60 005166  016200  0000006                    MOV     AT$DEV(R2),R0   ;Get physical device name
     61 005172  004767  0000006          2$:       CALL    CHKCLU          ;Convert CL and C1 names to unit numbers
     62 005176  103404                             BCS     10$             ;Br if not CL or C1 unit
     63 005200  010067  172660                     MOV     R0,TTXCL+4      ;Set CL unit number
     64 005204  012602                             MOV     (SP)+,R2
     65 005206  000207                             RETURN
     66 005210                          10$:       FABORT  #EM$CLX         ;Port unit must be CL or C1 unit
     67                                 ;
     68                                 ;  Process the /DTE=n SET HOST qualifier
     69                                 ;
     70 005220  010146                  HSTDTE:    MOV     R1,-(SP)
     71                                 ;
     72                                 ;  Accrue the CL unit number
     73                                 ;
     74 005222  004767  0000006                    CALL    ACRDEC          ;Accrue the CL unit number
     75 005226  010167  172632                     MOV     R1,TTXCL+4      ;Store CL unit index in EMT arg block
     76 005232  012601                             MOV     (SP)+,R1
     77 005234  000207                             RETURN
     78
     79         000001                             .END
Errors detected:  0

*** Assembler statistics

Work  file  reads:  0
Work  file writes:  0
Size of work file:  12016 Words  ( 47 Pages)
Size of core pool:  18176 Words  ( 71 Pages)
Operating  system:  RT-11

Elapsed time:  00:01:22.17
,LP:TSKST2=DK:TSKST2/C/N:SYM
```

Cross reference table (CREF V05.05)

Cross reference table (CREF V05.05)

Cross reference table (CREF V05.05)

Cross reference table (CREF V05.05)

```
CASTRO    1-98
CASTWO    1-98
CCLNAM    1-196
CCLSAV    1-94
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
CF$QUT    1-159     23-109     23-126
CFABLV    1-160
CFBLK     1-125
CFBUF     1-94
CFCHAN    1-124
CFEND     1-94
CFHOLD    1-132
CFIND     1-114
CFLFL4    1-119
CFNEST    1-117
CFPNT     1-125
CFSEND    1-118
CFSP      1-118
CFSPND    1-131
CFSTK     1-68
CFSTS     1-159     23-107     23-110     23-124     23-127
CHAIN     1-104
CHKALC    1-111     24-37
CHKBCM   25-11      25-22      25-33#
CHKCLU    1-36      33-61
CHKDEV    1-196
CHKDLM    1-186
CHKEQ     1-45       9-9       33-52
CHKMNT    1-171
CHKMTX    1-171
CHKTTD    1-194
CHNSIZ    1-41
CINDAT    1-141
CINFLG    1-48
CKACOJ    1-45      12-11      13-19
CKCLUS    1-37      33-27
CKPRIV    1-173     15-4
CKSYPV    1-45
CKTERM    1-174     14-40      16-4       17-4       18-7       19-4       20-5       21-4       22-8       22-16      23-150     23-157
         23-165     24-8       33-6
CL$COL    1-66
CL$EPN    1-41       5-99
CL$EPS    1-41
CL$LEN    1-101      5-115
CL$LIN    1-101     24-94*
CL$LIX    1-62      24-61      24-68      24-75      24-85      24-92      24-99      24-129     25-36      26-9       27-4       30-4
         31-4       33-23
```

Cross reference table (CREF V05.05)

```
CL$OPT    1-100     24-63*    24-70*
CL$RQH    1-63
CL$SKP    1-101      5-136
CL$WID    1-101      5-137     24-87*
CL$WQH    1-63
CL$XLN    1-37
CLAEMT    4-24#     28-13*    28-14*    28-15
CLCLOP    5-96       5-98      5-102     5-104     5-106     5-110     5-112     5-114     5-117     5-119     5-122     5-124
          5-127      5-128     5-129     5-132     24-68#
CLDEV     4-5#      32-21*    32-22
CLDEVX    1-139
CLEOFS    1-41      30-16
CLESTR    5-100     30-4#
CLFREE    1-57
CLLINE    1-57
CLLOOK   24-131     30-32     32-10#
CLNCKA    5-144#    24-34
CLOPHD    5-93#     24-31
CLOTIR    1-65
CLPRHD    5-130      5-151#   24-145
CLRDTR    5-17      18-6#
CLREMT    4-35#     25-24
CLRPRV    1-43       6-14
CLRTTB    5-14       5-19      5-21      5-23      5-25      5-27      5-34      5-37      5-47      5-49      5-52      5-53
          5-61       5-63      5-78      5-79     23-166#
CLRTTP    5-8        5-11      5-41      5-55      5-57      5-59     23-165#
CLSFCH    1-69      24-119
CLSFEP    1-42      30-36     30-36
CLSFRS    1-34      24-124
CLSFSP    1-98
CLSPFN   24-120     24-129#
CLTOP     5-133     24-92#
CLTOTL    1-98      24-9      33-18
CLTRNS    5-134     31-4#
CLUNIT    1-57
CLVERS    1-57      29-7*
CLXEMT    4-30#     25-13
CMDBUF    1-164      1-181
CMDCCL    1-196
CMDDSN    1-29
CMDFRM    1-29
CMDHD     1-28
CMDMEM    1-30
CMDOFF    1-28
CMDSET    1-30
CMDSHO    1-30
CMDUSE    1-30
CMDWHO    1-30
CO$8BT    1-115      5-125     5-126     5-127     5-128     5-129     5-132
CO$BNI    1-100      5-121     5-122
CO$BNO    1-100      5-123     5-124
CO$CR     1-99       5-95      5-96
CO$CTL    1-99       5-113     5-114
CO$DEF    1-66
CO$DTR    1-98       5-97      5-98
CO$FF     1-99       5-101     5-102     5-103     5-104
```

Cross reference table (CREF V05.05)

```
CO$FFO    1-99      5-105     5-106
CO$LC     1-99      5-109     5-110
CO$LFI    1-100     5-116     5-117
CO$LFO    1-100     5-118     5-119
CO$TAB    1-99      5-111     5-112
COAD      1-187     1-188
COAF      1-200
COAL      1-187     1-188
COLOO     1-181
COMMA     2-10#
CONFG2    1-62
CONFIG    1-113
CORUSR    1-61      6-10      10-15     11-22     13-14     23-20     23-99     23-105    23-116    23-122    33-30     33-36
CPFAUT    4-14#     5-174     11-29
CPFLAG    4-8#      6-13*      9-65*    11-29     13-8      13-25
CPFRES    4-13#     5-173     13-8      13-25
CPFSUS    4-12#     5-172     13-8
CPNAME    4-19#     6-11*      9-10      9-41      9-48      10-9      10-21
CPOPHD    5-167#    6-18
CPPFLG    5-172     5-173     5-174      9-65#
CPPID     4-17#     6-10*     7-18*     10-14     11-22     12-10     13-13
CPPNAM    5-170     9-4#
CPPPID    5-169     7-4#
CPPPRI    5-171     8-4#
CPPRIO    4-18#     6-12*     8-17*     12-8      12-16     12-17
CPSFLG    6-26      13-4#
CPSNAM    6-23      10-4#
CPSPRI    6-24      12-4#
CPSPRV    6-25      11-4#
CPUAH     1-181
CPUAL     1-181
CR        2-5#
CRLF      1-176      8-28
CS$RON    1-80
CSHALC    1-103
CSHDEV    1-108
CSHDVN    1-108
CSHHD     1-163
CSHMSG    1-203
CSHSIZ    1-143
CSIMS1    1-175     24-45
CSIMS2    1-173
CSIMS4    1-195
CTRLTT    1-102
CURMTX    1-186
CURPRM    1-130
CVDVNM    1-183
CVTTAB    1-164
CVTUC     1-163
CW$SOH    1-113
CW$PRO    1-62
CXTBAS    1-107
CXTPAG    1-79
CXTRMN    1-41
CXTWDS    1-107
DATTIM    1-29
```

Cross reference table (CREF V05.05)

```
DCCRD     1-137
DCCWR     1-137
DCTRD     1-137
DCTWR     1-137
DEADEV    1-171
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
DFJMEM    1-68
DIABFL    1-125     23-91
DIABLO    1-147     23-92
DIABNO    1-126     23-90
DIVIDE    1-180
DIVSOR    1-193
DJABMS    1-191
DKASHD    1-59
DKSAV     1-164
DLCEMT    1-30
DLMSG     1-191
DLTXT     1-178
DMTALL    1-191
DMTARG    1-170
DMTSUB    1-196
DOASGN    1-85
DORUN     1-29
DOSTOP    1-192
DTRCOM   18-5       18-7#
DTREMT    4-57#     18-4*      18-6*      18-9*      18-11
DVSHH1    1-59
DVSHH2    1-59
DVSHH3    1-59
DVSTAT    1-76
DZTXT     1-198
EDIT      1-73
EDTFIL    1-184
EM$ACL    1-54      28-31
EM$BYP    1-39      25-35
EM$CAP    1-44      11-63
EM$CIP    1-53
EM$CLB    1-54      28-37      33-32
EM$CLN    1-53      24-57      26-15      27-17      33-25
EM$CLU    1-34      15-22
EM$CLX    1-36      33-66
EM$CNO    1-44      10-34
EM$CPO    1-44      11-59
EM$CSE    1-70      24-49      31-40
EM$HNI    1-68
EM$ICL    1-52      21-15      27-15
EM$IDR    1-42      13-38
EM$ILN    1-53       1-54       7-27      14-36      16-20      17-11      18-15      28-28
EM$IST    1-42
EM$IUN    1-53      24-11      24-115     24-137     28-25      32-32      33-20
```

Cross reference table (CREF V05.05)

```
EM$IVN    1-53      29-9
EM$LAS    1-34      15-10
EM$LFD    1-34      15-16
EM$NAD    1-46
EM$NPD    1-53
EM$NPR    1-45      10-35
EM$NSF    1-53
EM$NSL    1-54
EM$NUK    1-86
EM$OPR    1-36
EM$PTA    1-36
EM$PTV    1-36
EM$SLT    1-54
EM$SLW    1-54
EM$SPL    1-39
EM$STL    1-42       9-55      30-47
EM$TMT    1-34      31-26      31-57
EM$TSL    1-54      28-34
EM$UIO    1-55
EM$WC0    1-40
EM$WC1    1-40
EM$WC2    1-40
EM$WC3    1-40
EM$WCM    1-40
ERRLOC    1-67      28-22
ERRSEV    1-134
ESC       1-82
FC$CDX    1-163
FC$LNK    1-163
FD$NAM    1-163
FF        2-9#
FILNAM    1-165     1-166
FKILL     1-165      7-27      7-28      10-34     10-35     11-59     11-63     13-38     14-36     15-10     15-16     15-22
         16-20      17-11     18-15     21-15     23-153    24-11     24-45     24-49     24-57     24-115    24-137    25-35
         26-15      27-15     27-17     28-25     28-28     28-31     28-34     28-37     29-9      30-47     31-26     31-40
         31-57      32-32     33-20     33-25     33-32     33-66
FPRINT    1-165      8-22      9-55
FSTDL     1-89
FSTIOL    1-62       1-79
GAGMSG    1-190
GENTOP    1-103
GETKCH    1-77
GRT1      1-144
GTRD50    1-170     33-53
H.CSR     1-76
H.VEC     1-76
HANBSY    1-175
HANCHN    1-64
HANENT    1-75
HANPAR    1-76
HANSIZ    1-75
HAZEL     1-77      23-75
HAZLFL    1-77      23-74
HAZLNO    1-77      23-73
HIMAP     1-142
HIPRI     1-176
```

Cross reference table (CREF V05.05)

```
HNBUF     1-174
HOSTHD    5-159#    33-11
HSTDTE    5-160      5-161     33-70#
HSTPRT    5-162     33-51#
HUPARG    1-195
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      14-49     23-32*
IN$ACT    1-157
IN$CMD    1-157
IN$CNT    1-157
INDACT    1-167
INDERR    1-106
INDSAV    1-157
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
INVOPT    1-165      1-171     1-186
INVTIM    1-191
IOABFL    1-61
ITRMTP    1-159     23-28*
JCXPGS    1-142
JCXSMS    1-198
JPWDEV    1-39
JPWFLG    1-39
JPWTYP    1-39
JSTKND    1-102
JSWLOC    1-67
K52       1-73
KBMSG     1-180
KBTX      1-184
KCSIBF    1-169
KCSIMS    1-170
KDOCIN    1-28
KED       1-73
KILEMT    1-192
KL3CLR    1-87
KL4CLR    1-127
KMNBAS    1-154
KMNCHN    1-94
KMNHI     1-78
KMNNAM    1-196
KMNPGS    1-79
```

Cross reference table (CREF V05.05)

```
KMNSTK    1-79
KMNSTR    1-79
KMNTOP    1-79
KMPRMT    1-138
L         5-5       5-5#      5-6       5-6#      5-7       5-7#      5-8       5-8#      5-9       5-9#      5-10      5-10#
          5-11      5-11#     5-12      5-12#     5-13      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#     5-20      5-20#     5-21      5-21#     5-22      5-22#
          5-23      5-23#     5-24      5-24#     5-25      5-25#     5-26      5-26#     5-27      5-27#     5-28      5-28#
          5-29      5-29#     5-30      5-30#     5-31      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#     5-40      5-40#
          5-41      5-41#     5-42      5-42#     5-43      5-43#     5-44      5-44#     5-45      5-45#     5-46      5-46#
          5-47      5-47#     5-48      5-48#     5-49      5-49#     5-50      5-50#     5-51      5-51#     5-52      5-52#
          5-53      5-53#     5-54      5-54#     5-55      5-55#     5-56      5-56#     5-57      5-57#     5-58      5-58#
          5-59      5-59#     5-60      5-60#     5-61      5-61#     5-62      5-62#     5-63      5-63#     5-64      5-64#
          5-65      5-65#     5-66      5-66#     5-67      5-67#     5-68      5-68#     5-69      5-69#     5-70      5-70#
          5-71      5-71#     5-72      5-72#     5-73      5-73#     5-74      5-74#     5-75      5-75#     5-76      5-76#
          5-77      5-77#     5-78      5-78#     5-79      5-79#     5-85      5-85#     5-86      5-86#     5-87      5-87#
          5-94      5-94#     5-95      5-95#     5-96      5-96#     5-97      5-97#     5-98      5-98#     5-99      5-99#
          5-100     5-100#    5-101     5-101#    5-102     5-102#    5-103     5-103#    5-104     5-104#    5-105     5-105#
          5-106     5-106#    5-107     5-107#    5-108     5-108#    5-109     5-109#    5-110     5-110#    5-111     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#
          5-124     5-124#    5-125     5-125#    5-126     5-126#    5-127     5-127#    5-128     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#    5-145     5-145#
          5-146     5-146#    5-152     5-152#    5-153     5-153#    5-154     5-154#    5-160     5-160#    5-161     5-161#
          5-162     5-162#    5-168     5-168#    5-169     5-169#    5-170     5-170#    5-171     5-171#    5-172     5-172#
          5-173     5-173#    5-174     5-174#
LA120     1-147     23-39
LA12FL    1-148     23-38
LA12NO    1-148     23-37
LA36      1-147     23-15
LA36FL    1-126     23-14
LA36NO    1-126     23-13
LACTIV    1-80
LAFSIZ    1-83
LBSPRI    1-160     12-16*
LCBIT     1-147
LCDTYP    1-66
LCLTXT    1-57
LCLUNT    1-62      15-20
LCOL      1-82       1-133
LCONTM    1-91
LCPUHI    1-91
LCPULO    1-91
LCXTBL    1-34      31-24
LD$RON    1-121
LDBASE    1-122
LDCLEN    1-71
LDDEVX    1-124
LDFLAG    1-122
LDMNT     1-70
LDNAM     1-169
LDNAME    1-122
LDOPHD    1-177
LDPDEV    1-122
```

Cross reference table (CREF V05.05)

```
LDSIZE   1-122
LF       2-4#
LF$OPN   1-162
LF$WRT   1-162
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    16-7      21-5      22-9      22-17     24-102    26-11     27-7
LNAME    1-60
LNBLKS   1-107
LNMAP    1-113
LNPRIM   1-113    22-25
LNSBLK   1-108
LNSPAC   1-117
LOCKTX   1-180
LOFSPC   1-110
LOGASN   1-172
LOGBAS   1-132     1-134
LOGBLK   1-161
LOGBUF   1-161
LOGCHK   1-133
LOGCHN   1-161
LOGCLS   1-176
LOGDVU   1-132     1-134
LOGFLG   1-161
LOGPTR   1-161
LOMAP    1-142
LOTBUF   1-81
LOTNXT   1-81
LOTPNT   1-81
LOTSIZ   1-82
LOTSPC   1-82
LOUTIR   1-65
LP$7BT   1-52     21-8      21-11     27-8      27-11
LP$ODD   1-52      5-86      5-153    22-10     22-18     26-12
LP$PAR   1-52      5-85      5-86      5-152     5-153    22-10     22-18     26-12
LP$SPD   1-52     16-8      24-103
LPRG1    1-145    33-37*    33-44*
LPRG2    1-145    33-38*    33-45*
LPRI     1-160    12-17*
LPROG    1-90
LPROJ    1-90
LRBFIL   1-114
LRDTIM   1-80
LSCCA    1-110
```

Cross reference table (CREF V05.05)

```
LSECPT    1-95
LSPEMT    4-46#     16-10*    16-11*    16-12*    16-13     24-105*   24-107*   24-108
LSTACT    1-78
LSTATE    1-140
LSTDL     1-89
LSTHL     1-62
LSTIOL    1-62
LSTMX     1-156
LSTPL     1-135     14-34
LSTPRM    1-131
LSTSL     1-140      7-14
LSTSPL    1-76
LSUCF     1-86
LSW       1-61       7-16      14-47     15-8      19-6      23-151
LSW2      1-104      5-6       5-8       5-13      5-14      5-18      5-19      5-20      5-21      5-22      5-23      5-33
          5-34       5-36      5-37      5-40      5-41      5-45      5-47      5-52      5-53      5-54      5-55      5-56
          5-57       5-58      5-59      5-60      5-61      5-62      5-63      5-77      5-78      5-79      14-49*    16-6*
         19-5*      23-6*     23-7*     23-13*    23-14*    23-29     23-32     23-37*    23-38*    23-45*    23-46*    23-52*
         23-53*     23-59*    23-60*    23-66*    23-67*    23-73*    23-74*    23-81*    23-82*    23-90*    23-91*    23-103*
         23-120*    23-132*
LSW2S     1-109     23-29*
LSW3      1-109      5-10      5-11      15-14
LSW4      1-127      5-24      5-25      23-104*   23-121*
LSW5      1-87      23-139*   23-145*
LSW6      1-155      5-48      5-49      23-138*   23-144*
LSW7      1-159      5-26      5-27      23-16*    23-40*    23-76*    23-84*    23-93*    23-133*
LSW8      1-123
LSW9      1-85
LTRMTP    1-148     23-8*     23-15*    23-28     23-39*    23-47*    23-54*    23-61*    23-68*    23-75*    23-83*    23-92*
LTSCMD    1-117
LUNAME    1-90      10-20
LWINDO    1-34      23-22
MAXALC    1-63
MAXASN    1-106
MAXAVL    1-174
MAXMEM    1-67
MAXMTX    1-186
MAXPRI    1-67       1-160
MAXSEC    1-95
MDT       1-77
MHNSIZ    1-96
MHNSMS    1-97
MINTIM    1-95
MISSEQ    1-175
MNBASE    1-185
MNBPC     1-184
MNFLGS    1-184
MNTARG    1-195
MNTDEV    1-170
MNTFUL    1-172
MNTOP     1-185
MNTTXT    1-197
MONAR1    1-185
MONAR2    1-185
MONHD     1-185
MONVEC    1-160     23-107    23-110    23-124    23-127
```

Cross reference table (CREF V05.05)

```
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    8-15    8-26
MXPRMT   1-138
MXTTCT   1-34    31-55
MXVEC    1-158
NAMTOP   1-144
NARGS    5-4#    5-5     5-5     5-6     5-6     5-7     5-7     5-8     5-8     5-9     5-9     5-10
         5-10    5-11    5-11    5-12    5-12    5-13    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    5-20    5-20    5-21    5-21    5-22
         5-22    5-23    5-23    5-24    5-24    5-25    5-25    5-26    5-26    5-27    5-27    5-28
         5-28    5-29    5-29    5-30    5-30    5-31    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    5-40
         5-40    5-41    5-41    5-42    5-42    5-43    5-43    5-44    5-44    5-45    5-45    5-46
         5-46    5-47    5-47    5-48    5-48    5-49    5-49    5-50    5-50    5-51    5-51    5-52
         5-52    5-53    5-53    5-54    5-54    5-55    5-55    5-56    5-56    5-57    5-57    5-58
         5-58    5-59    5-59    5-60    5-60    5-61    5-61    5-62    5-62    5-63    5-63    5-64
         5-64    5-65    5-65    5-66    5-66    5-67    5-67    5-68    5-68    5-69    5-69    5-70
         5-70    5-71    5-71    5-72    5-72    5-73    5-73    5-74    5-74    5-75    5-75    5-76
         5-76    5-77    5-77    5-78    5-78    5-79    5-79    5-84#   5-85    5-85    5-86    5-86
         5-87    5-87    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    5-100   5-100   5-101   5-101   5-102   5-102   5-103   5-103   5-104
         5-104   5-105   5-105   5-106   5-106   5-107   5-107   5-108   5-108   5-109   5-109   5-110
         5-110   5-111   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   5-124   5-124   5-125   5-125   5-126   5-126   5-127   5-127   5-128
         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   5-145   5-145   5-146   5-146   5-151#  5-152   5-152   5-153   5-153   5-154   5-154
         5-159#  5-160   5-160   5-161   5-161   5-162   5-162   5-167#  5-168   5-168   5-169   5-169
         5-170   5-170   5-171   5-171   5-172   5-172   5-173   5-173   5-174   5-174
NATXT    1-178
NEDCHR   1-65
NEWJSW   1-102
NFRESB   1-129
NINTXT   1-178
NLINES   1-115
NOCCL    1-177
NOCIN    1-169
NODAT    1-183
NOFIL    1-181    1-189
NOFLAG   1-171
NOFRDL   1-200
NOIND    1-175
NOKMON   1-196
NOLDMT   1-183
NONEMS   1-183
NOOPTT   1-189
NOPMGN   1-185
NOPRG    1-165    1-169
```

Cross reference table (CREF V05.05)

```
NOSTRT    1-168
NOTAVL    1-184
NOTON     1-190       7-28
NOTXT     1-178
NOUDC     1-182
NSPLDV    1-121
NSWPMS    1-186
NUCHN     1-114
NUMDCD    1-138
NUMDEV    1-151
NUMON     1-83
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        6-19
OT$RON    1-153
OTHRON    1-194
OTRMNT    1-196
OVRCOR    1-168
PO$BYP    1-48        25-33
PO$DBG    1-47
PO$NAM    1-45        10-17
PO$OPR    1-45
PO$SPV    1-43
P2$CGR    1-40
P2$TRM    1-45
PA$BEL    1-36
PA$BLD    1-35
PA$DSC    1-35
PA$DWD    1-35
PA$GRC    1-35
PA$HQL    1-35
PA$LET    1-35
PA$NWD    1-36
PA$UKC    1-35
PA$ULN    1-35
PARHD     5-84#       20-10
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         4-71        11-11
PFSO      1-43         4-64        11-10
```

Cross reference table (CREF V05.05)

```
PHYMEM    1-101
PMBUSY    1-185
PNAME     1-151      1-174
POPCF     1-175
PRGALL    1-29
PRGSIZ    1-78
PRGTOP    1-78
PRIVAO    1-44
PRIVCO    1-47       10-17      25-33
PRIVC2    1-45
PRIVSO    1-43
PRMBUF    1-131
PRMEND    1-131
PRMPNT    1-130
PROSLT    1-52
PRSTRN   22-26       31-7       31-19#
PRTBUF    1-183
PRTDAT    1-199
PRTDC2    1-180
PRTDC3    1-180
PRTDEC    1-174      8-24       8-27
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
PRVLST    1-47
PRVOPT    1-43       5-168
PUSHCF    1-166
PVCEMT    4-69#      11-33*     11-45
PVNPW     1-43       11-12
PVON      1-87
PVSEMT    4-62#      11-32*     11-38
QHDMS1    1-59
QHDMS2    1-59
QUME      1-147
QUMEFL    1-149
QUMENO    1-149
R$CHN     1-41
R$XCHN    1-41
R50BUF    1-170      24-36      33-57
R50C10    4-7#       32-18
R50CLO    4-6#       32-20
R50COM    1-110
R50DIR    1-166
R50DK     1-199
R50DSK    1-172
R50DUP    1-167
```

Cross reference table (CREF V05.05)

```
R50HST    4-4#      33-37     33-38
R50IND    1-166
R50K52    1-167
R50KED    1-167
R50KEX    1-167
R50KMN    1-37      33-44     33-45
R50LD     1-172
R50LD0    1-170
R50LD7    1-172
R50LOG    1-176
R50MON    1-200
R50NO     1-173
R50PIP    1-167
R50SAV    1-166
R50SY     1-166
R50TSX    1-167
R50TT     1-195
R50UCL    1-167
R50VIR    1-168
RC$$SZ    1-51
RC$AEP    1-50
RC$AGE    1-50
RC$CNT    1-49
RC$EXC    1-50
RC$FLG    1-50
RC$GBL    1-50
RC$LCG    1-52
RC$LEN    1-51
RC$NAM    1-50
RC$OWN    1-49
RC$PVT    1-51
RC$USE    1-50
RCBBAS    1-51
RCBEND    1-51
RDB       1-139
RDBEND    1-139
RDCMD     1-164     5-29      5-30      5-35      6-30      14-60     23-101    23-118    24-53     33-47
RDERM     1-168
REMTXT    1-57
RESDEV    1-153
RJEMT     1-37      13-24
RNMS      1-179
RONTXT    1-184
RSR       1-156
RSTPRV    1-44
RSTQUT    5-43      23-117    23-120#
RT$$SZ    1-139
RT$NAM    1-139
RUNDEV    1-107
RUNEMT    1-168
RUNHD     1-164
RUNMS     1-200     23-153
S$HICP    1-92
S$INWT    1-93
S$IOFN    1-88
S$IOWT    1-145
```

Cross reference table (CREF V05.05)

```
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       19-8
SADM3A    5-5        23-81#
SAUTO     5-7        19-4#
SBPSUF    1-36
SCHAIN    1-124
SCNOPS    1-37       14-56      33-12
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
SDIAB     5-15       23-90#
SDNAME    1-140
SDSFCB    1-119
SDSKIP    1-127
SEARCH    1-164      24-32
SERFLG    1-61
SETCL     1-26       24-8#
SETDED    5-10       23-150#
SETDTR    5-16       18-4#
SETHD     1-173
SETHST    1-26       33-6#
SETJMP   23-31       23-33#
SETNSC    5-46       23-132#
SETPAR    5-38       20-4#
SETPRC    1-27        6-6#
SETPRM    4-16#      14-9*      14-41*     23-26      23-30
SETQUT    5-42       23-100     23-103#
SETTNW    5-75       23-144#
```

Cross reference table (CREF V05.05)

```
SETTTB   5-6     5-13    5-18    5-20    5-22    5-24    5-26    5-33    5-36    5-45    5-48    5-60
         5-62    5-77    23-152  23-158#
SETTTP   5-40    5-54    5-56    5-58    23-157#
SETTTY   1-27    14-5#
SETTW    5-74    23-138#
SETWRD   1-42    20-11   24-146
SETXON   5-76    17-4#
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
SH$FLG   1-74
SH$NAM   1-74
SH$RTN   1-74
SH$VAL   1-74
SHMTH1   1-182
SHMTH2   1-182
SHOHD    1-197
SHRRCB   1-51
SHRRCN   1-51
SHTMSG   1-192
SIZEMT   1-195
SIZVAL   1-169
SJEMT    1-37    13-27
SKPSPC   1-28     9-14   14-10   20-6    24-17   24-25   24-141  31-30   31-37   31-48   31-70
SLA120   5-31    23-37#
SLKDOF   1-55
SLKDON   1-55
SLMXLN   1-203
SMRSIZ   1-143
SO$NO    1-75
SO$NVL   1-75
SO$OCT   1-75
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
```

Cross reference table (CREF V05.05)

```
SPLHD     1-186
SPLHLA    1-177
SPLPND    1-194
SPSNG     1-187     1-189
SPUBUF    1-69
SPWFM     1-187     1-188
SQUME     5-44     23-89#
SRTSIZ    1-143
SRTSMS    1-194
SRTTXT    1-197
SSRMAP    1-197
STAEMT    4-76#     15-26*    15-27
STCLBT    5-94     27-4#
STCLEN    5-9      21-4#
STCLGR    5-107     5-108    24-85#
STCLLN    5-120    28-4#
STCLNL   24-57#    24-62     24-69     24-76     24-86     24-93     24-100    24-130    25-38     30-6      31-6
STCLNP    5-131    26-19#
STCLOP    5-95      5-97      5-101     5-103     5-105     5-109     5-111     5-113     5-116     5-118     5-121     5-123
          5-125     5-126    24-61#
STCLP1   26-9#     26-20
STCLPO    5-130    24-141#
STCLPR    5-152     5-153     5-154    26-8#
STCLRB    5-146    25-22#
STCLRS    5-135    24-124#
STCLS1   24-105#    26-14     27-16
STCLSP    5-138    24-99#
STCLVL    5-99      5-115     5-136     5-137    24-75#
STCLVR    5-139    29-4#
STCLXB    5-145    25-11#
STCLXN    5-140    24-119#
STERM    23-48#
STHAZL    5-28     23-73#
STLA36    5-12      5-32     23-13#
STLGCN    1-29
STLGHD    1-176
STNOPR    5-39     22-16#
STNOVR    1-27     23-99#
STPAR     5-85      5-86      5-87     22-8#
STPASK    1-194
STPFLG    1-92
STSP1    16-10#    19-9      21-17     22-12     22-19
STTRNS    5-64     22-24#
STTTSP    5-50     16-4#
STTTST    5-51     15-4#
STVRFY    1-27     23-116#
SUBARG    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
```

Cross reference table (CREF V05.05)

```
SUMS      1-144
SUPCOD    1-144
SVT100    1-27        5-65        23-45#
SVT200    1-27        5-66         5-67         5-70         5-71         23-52#
SVT227    1-26        5-68        23-59#
SVT228    1-26        5-69        23-66#
SVT50     1-27        5-72         5-73        23-6#
SWPTX     1-180
SXBPNT    1-69
SYASHD    1-59
SYHD1     1-179
SYHD2     1-179
SYINDX    1-151
SYNAME    1-152
SYPSWD    1-39
SYSAV     1-164
SYSDAT    1-141
SYTIMH    1-141
SYTIML    1-141
SYUNIT    1-151
TAB       2-8#
TALEMT    1-114
TBLOVF    1-173
TECO      1-73
TK1SEC    1-143
TK1VAL    1-141
TM$AUT    1-57
TM$CDS    1-56
TM$CEN    1-56
TM$CL0    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$NAD    1-60
TM$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        33-46
TMIDLH    1-71
```

Cross reference table (CREF V05.05)

```
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       8-25
TRGRET    1-144
TRMHD1    1-56
TRMHD2    1-56
TRMSTR    1-166
TSKST2    1-5#        1-28
TSR       1-156
TSXLN     1-144
TSXSIT    1-144
TSXSMS    1-198
TTHD      5-4#        14-55
TTXCL     4-40#       33-10*      33-17       33-22*      33-39       33-63*      33-75*
TXTCL     1-91
UC$MDC    1-163
UC$NDC    1-163
UCHAN     1-112
UCIDEF    1-64
UCISPC    1-96
UCLBLK    1-162
UCLCMD    1-28
UCLDAT    1-162
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
VCSHNB    1-95
VDBFLG    1-82
VHIPCT    1-136
VIMAGE    1-102
VINTIO    1-128
VLDSYS    1-86
VNUMDC    1-138
VPRIDF    1-42
VQUAN0    1-135       1-136
VQUAN1    1-136
VQUAN2    1-136
```

Cross reference table (CREF V05.05)

```
VQUAN3    1-135     1-136
VQUN1A    1-136
VQUN1B    1-128
VQUN1C    1-128
VSLEDT    1-203
VT100     1-147     23-47
VT10FL    1-149     23-46
VT10NO    1-149     23-45
VT200     1-146     23-54
VT2007    1-146     23-61
VT2008    1-146     23-68
VT20FL    1-150     23-53     23-60     23-67
VT20NO    1-150     23-52     23-59     23-66
VT52      1-147     23-8
VT52FL    1-148     23-7
VT52NO    1-126     23-6
WILDFL    1-72
WINSTT    4-82#     23-24
WLDNAM    2-12#
XAREA     1-165     23-107    23-110    23-124    23-127    24-132    30-36     32-22
XONEMT    4-52#     17-5*     17-6*     17-7
YESTXT    1-178
ZCLR      1-156
```

Cross reference table (CREF V05.05)

```
...CM1   24-132    30-36    32-22
...CM2   23-107    23-110   23-110   23-124   23-127   23-127   24-132   24-132   24-132   24-132   24-132   24-132
         30-36     30-36    30-36    30-36    30-36    32-22    32-22
...CM3   24-134    24-136   30-37
...CM5    8-25      8-28    23-107   23-110   23-124   23-127   24-132   30-36    32-22    33-46
...CM6   23-107    23-110   23-124   23-127
.CLOSE    1-20#    24-134   30-37
.CSIGE    1-18#
.CSISP    1-16#
.DATE     1-19#
.ENTER    1-21#
.EXIT     1-21#
.FPROT    1-22#
.GTIM     1-19#
.GTLIN    1-19#
.GVAL     1-22#    23-107   23-124
.HERR     1-22#
.LOOKU    1-20#    32-22
.PRINT    1-20#     8-25     8-28    33-46
.PURGE    1-17#    24-136
.PVAL     1-22#    23-110   23-127
.READW    1-17#
.REOPE    1-18#
.SAVES    1-18#
.SERR     1-22#
.SPFUN    1-19#    24-132   30-36
.SRESE    1-16#
.TTOUT    1-16#
.TTYIN    1-17#
.TTYOU    1-17#
.WRITW    1-21#
CMDDEF    3-47#     5-5      5-6      5-7      5-8      5-9      5-10     5-11     5-12     5-13     5-14     5-15
          5-16      5-17     5-18     5-19     5-20     5-21     5-22     5-23     5-24     5-25     5-26     5-27
          5-28      5-29     5-30     5-31     5-32     5-33     5-34     5-35     5-36     5-37     5-38     5-39
          5-40      5-41     5-42     5-43     5-44     5-45     5-46     5-47     5-48     5-49     5-50     5-51
          5-52      5-53     5-54     5-55     5-56     5-57     5-58     5-59     5-60     5-61     5-62     5-63
          5-64      5-65     5-66     5-67     5-68     5-69     5-70     5-71     5-72     5-73     5-74     5-75
          5-76      5-77     5-78     5-79     5-85     5-86     5-87     5-94     5-95     5-96     5-97     5-98
          5-99      5-100    5-101    5-102    5-103    5-104    5-105    5-106    5-107    5-108    5-109    5-110
          5-111     5-112    5-113    5-114    5-115    5-116    5-117    5-118    5-119    5-120    5-121    5-122
          5-123     5-124    5-125    5-126    5-127    5-128    5-129    5-130    5-131    5-132    5-133    5-134
          5-135     5-136    5-137    5-138    5-139    5-140    5-145    5-146    5-152    5-153    5-154    5-160
          5-161     5-162    5-168    5-169    5-170    5-171    5-172    5-173    5-174
FABORT    3-15#     7-27     7-28    10-34    10-35    11-59    11-63    13-38    14-36    15-10    15-16    15-22
         16-20     17-11    18-15    21-15    23-153   24-11    24-45    24-49    24-57    24-115   24-137   25-35
         26-15     27-15    27-17    28-25    28-28    28-31    28-34    28-37    29-9     30-47    31-26    31-40
         31-57     32-32    33-20    33-25    33-32    33-66
FERR      3-4#      8-22     9-55
FWARN     3-23#
TBLDEF    3-35#     5-4      5-84     5-93     5-151    5-159    5-167
TBLEND    3-61#     5-80     5-88     5-147    5-155    5-163    5-175
```


TTTTT SSS K   K SSS H   H 000