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.
                                ;   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
                                        .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
                                        .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$$SZ,EM$NAD
                                        .GLOBL  INSTBN,AF$SCA,AF$NOW,AF$MEM,PO$DBG,PRIVCO,PRVLST
                                        .GLOBL  ABRTAD,ABRTCD,CINFLG,$VNOTT,II$PRV,II$NPV
                                        .GLOBL  TM$RD1,TM$RD2,TM$LCL,TM$GBL,SPACE1,RC$OWN,RC$CNT
                                        .GLOBL  RC$EXC,RC$AGE,RC$AEP,RC$USE,RC$FLG,RC$GBL,RC$NAM
                                        .GLOBL  RC$LEN,RC$PVT,RCBBAS,RCBEND,RC$$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,$8BIT,CHKALC
        .GLOBL  UCHAN,$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$8BT
	.GLOBL	CD$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	$WILD,ERRSEV,UERSEV,PASLIN,LOGBAS,LOGDVU
	.GLOBL	LSTPL,SDCB,SDCBND,VQUANO,VQUAN3
	.GLOBL	VQUAN1,VQUN1A,VQUAN2,VHIPCT,VQUANO,VQUAN3
	.GLOBL	DCTRD,DCCRD,DCTWR,DCCWR,ASNSRC
	.GLOBL	VCORTM,NUMDCD,VNUMDC,KMPRMT,MXPRMT
	.GLOBL	RDB,RDBEND,RT$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  SUBARO,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 cross connect our time-sharing line with a CL unit
                                         ;
000050     000     126                   TTXCL:  .BYTE   0,126
000052  000013                                   .WORD   13
000054  000000                                   .WORD   0
                                         ;
                                         ;  Emt to set line speed
                                         ;
000056     000     154                   LSPEMT: .BYTE   0,154
000060  000000                                   .WORD   0               ;Line number
000062  000000                                   .WORD   0               ;Speed code
                                         ;
                                         ;  Emt to reset XOFF status for a line
                                         ;
000064     001     154                   XONEMT: .BYTE   1,154
000066  000000                                   .WORD   0
                                         ;
                                         ;  EMT to set/reset DTR for a line
                                         ;
000070     002     154                   DTREMT: .BYTE   2,154           ;Assume raise DTR (3,154 = drop DTR)
000072  000000                                   .WORD   0
                                         ;
                                         ;  Emt to set privilege flags
                                         ;
000074     001     150                   PVSEMT: .BYTE   1,150
000076     002     001                           .BYTE   2,1
000100  000000G                                  .WORD   PFSO
000102  000000                                   .WORD   0
                                         ;
```

```
                                     ;  Emt to clear privilege flags
                                     ;
   000104      001      150          PVCEMT: .BYTE   1,150
   000106      001      001                  .BYTE   1,1
   000110   000000G                          .WORD   PFCO
   000112   000000                           .WORD   0
                                     ;
                                     ;  Emt to start an inactive line
                                     ;
   000114      000      126          STAEMT: .BYTE   0,126
   000116   000020                           .WORD   20
   000120   000000                           .WORD   0
                                     ;
                                     ;  Emt to declare terminal type change to process windowing system
                                     ;
   000122      007      161          WINSTT: .BYTE   7,161
```

```
                          ;-------------------------------------------------------------
                          ;  Options for SET TT command
                          ;
000124                              TBLDEF  TT,3
000002                              CMDDEF  ADM*3A,SADM3A,0,0
000012                              CMDDEF  AUTO*BAUD,SAUTO,0,0
000022                              CMDDEF  NOAUTO*BAUD,CLRTTP,LSW2,$AUTO
000032                              CMDDEF  BIT*S,STCLEN,0,0
000042                              CMDDEF  DEAD,SETDED,LSW3,$DEAD
000052                              CMDDEF  NODEAD,CLRTTP,LSW3,$DEAD
000062                              CMDDEF  DEC*WRITER,STLA36,0,0
000072                              CMDDEF  DEF*ER,SETTTB,LSW2,$DEFER
000102                              CMDDEF  NOD*EFER,CLRTTB,LSW2,$DEFER
000112                              CMDDEF  DI*ABLO,SDIAB,0,0
000122                              CMDDEF  DT*R,SETDTR,0,0
000132                              CMDDEF  NODT*R,CLRDTR,0,0
000142                              CMDDEF  E*CHO,SETTTB,LSW2,$ECHO
000152                              CMDDEF  NOE*CHO,CLRTTB,LSW2,$ECHO
000162                              CMDDEF  EIGHT*BIT,SETTTB,LSW2,$8BIT
000172                              CMDDEF  NOEIGHT*BIT,CLRTTB,LSW2,$8BIT
000202                              CMDDEF  FORM,SETTTB,LSW2,$FORM
000212                              CMDDEF  NOFORM,CLRTTB,LSW2,$FORM
000222                              CMDDEF  FORM0,SETTTB,LSW4,$FORM0
000232                              CMDDEF  NOFORM0,CLRTTB,LSW4,$FORM0
000242                              CMDDEF  G*AG,SETTTB,LSW7,$TTGAG
000252                              CMDDEF  NOG*AG,CLRTTB,LSW7,$TTGAG
000262                              CMDDEF  HAZEL*TINE,STHAZL,0,0
000272                              CMDDEF  HO*LD,RDCMD,0,0
000302                              CMDDEF  NOH*OLD,RDCMD,0,0
000312                              CMDDEF  LA1*20,SLA120,0,0
000322                              CMDDEF  LA3*6,STLA36,0,0
000332                              CMDDEF  LC,SETTTB,LSW2,$LC
000342                              CMDDEF  NOL*C,CLRTTB,LSW2,$LC
000352                              CMDDEF  LE*NGTH,RDCMD,0,0
000362                              CMDDEF  PAG*E,SETTTB,LSW2,$PAGE
000372                              CMDDEF  NOP*AGE,CLRTTB,LSW2,$PAGE
000402                              CMDDEF  PAR*ITY,SETPAR,0,0
000412                              CMDDEF  NOPAR*ITY,STNOPR,0,0
000422                              CMDDEF  PHO*NE,SETTTP,LSW2,$PHONE
000432                              CMDDEF  NOPHO*NE,CLRTTP,LSW2,$PHONE
000442                              CMDDEF  QUI*ET,SETQUT,0,0
000452                              CMDDEF  NOQ*UIET,RSTQUT,0,0
000462                              CMDDEF  QUM*E,SQUME,0,0
000472                              CMDDEF  SC*OPE,SETTTB,LSW2,$SCOPE
000502                              CMDDEF  NOSC*OPE,SETNSC,0,0
000512                              CMDDEF  SEVEN*BIT,CLRTTB,LSW2,$8BIT
000522                              CMDDEF  SI*NGLE,SETTTB,LSW6,$STSNG
000532                              CMDDEF  NOSI*NGLE,CLRTTB,LSW6,$STSNG
000542                              CMDDEF  SPE*ED,STTTSP,0,0
000552                              CMDDEF  STA*RT,STTTST,0,0
000562                              CMDDEF  SYSP*ASSWORD,SETTTP,LSW2,$SYSPS
000572                              CMDDEF  NOSYSP*ASSWORD,CLRTTP,LSW2,$SYSPS
000602                              CMDDEF  SYSPS,SETTTP,LSW2,$SYSPS
000612                              CMDDEF  NOSYSPS,CLRTTP,LSW2,$SYSPS
000622                              CMDDEF  SYSP*WD,SETTTP,LSW2,$SYSPS
000632                              CMDDEF  NOSYSP*WD,CLRTTP,LSW2,$SYSPS
000642                              CMDDEF  TAB,SETTTB,LSW2,$TAB
```

```
000652                                  CMDDEF  NOTAB,CLRTTB,LSW2,$TAB
000662                                  CMDDEF  TAP*E,SETTTB,LSW2,$TAPE
000672                                  CMDDEF  NOTAP*E,CLRTTB,LSW2,$TAPE
000702                                  CMDDEF  TR*ANSLATE,STTRNS,0,0
000712                                  CMDDEF  VT1*00,SVT100,24.,0
000722                                  CMDDEF  VT2*00,SVT200,24.,0
000732                                  CMDDEF  VT220,SVT200,24.,0
000742                                  CMDDEF  VT240,SVT200,24.,0
000752                                  CMDDEF  VT241,SVT200,24.,0
000762                                  CMDDEF  VT5*0,SVT50,12.,0
000772                                  CMDDEF  VT5*2,SVT50,24.,0
001002                                  CMDDEF  W*AIT,SETTW,0,0
001012                                  CMDDEF  NOW*AIT,SETTNW,0,0
001022                                  CMDDEF  XON,SETXON,0,0
001032                                  CMDDEF  8BIT,SETTTB,LSW2,$8BIT
001042                                  CMDDEF  NO8BIT,CLRTTB,LSW2,$8BIT
001052                                  CMDDEF  7BIT,CLRTTB,LSW2,$8BIT
001062                                  TBLEND
                              ;
                              ;  Define options for SET TT [n] PARITY= command
                              ;
000124                                  TBLDEF  PAR,3
001066                                  CMDDEF  E*VEN,STPAR,LP$PAR
001076                                  CMDDEF  O*DD,STPAR,LP$PAR!LP$ODD
001106                                  CMDDEF  N*ONE,STPAR,0
001116                                  TBLEND
                              ;
                              ;
                              ;  Define options for the SET CLn command
                              ;
000124                                  TBLDEF  CLOP,2
001122                                  CMDDEF  BIT*S,STCLBT,0
001130                                  CMDDEF  CR,STCLOP,CO$CR
001136                                  CMDDEF  NOCR,CLCLOP,CO$CR
001144                                  CMDDEF  DTR,STCLOP,CO$DTR
001152                                  CMDDEF  NODTR,CLCLOP,CO$DTR
001160                                  CMDDEF  ENDP*AGES,STCLVL,CL$EPN
001166                                  CMDDEF  ENDS*TRING,CLESTR,0
001174                                  CMDDEF  FORM,STCLOP,CO$FF
001202                                  CMDDEF  NOFORM,CLCLOP,CO$FF
001210                                  CMDDEF  FF,STCLOP,CO$FF
001216                                  CMDDEF  NOFF,CLCLOP,CO$FF
001224                                  CMDDEF  FORM0,STCLOP,CO$FF0
001232                                  CMDDEF  NOFORM0,CLCLOP,CO$FF0
001240                                  CMDDEF  GR*APH,STCLGR,0.
001246                                  CMDDEF  NOGR*APH,STCLGR,132.
001254                                  CMDDEF  LC,STCLOP,CO$LC
001262                                  CMDDEF  NOLC,CLCLOP,CO$LC
001270                                  CMDDEF  TAB,STCLOP,CO$TAB
001276                                  CMDDEF  NOTAB,CLCLOP,CO$TAB
001304                                  CMDDEF  CTRL,STCLOP,CO$CTL
001312                                  CMDDEF  NOCTRL,CLCLOP,CO$CTL
001320                                  CMDDEF  LEN*GTH,STCLVL,CL$LEN
001326                                  CMDDEF  LFIN,STCLOP,CO$LFI
001334                                  CMDDEF  NOLFIN,CLCLOP,CO$LFI
001342                                  CMDDEF  LFOUT,STCLOP,CO$LFO
001350                                  CMDDEF  NOLFOUT,CLCLOP,CO$LFO
```

```
001356                          CMDDEF  LIN*E,STCLLN,0
001364                          CMDDEF  BININ,STCLOP,CO$BNI
001372                          CMDDEF  NOBININ,CLCLOP,CO$BNI
001400                          CMDDEF  BINOUT,STCLOP,CO$BNO
001406                          CMDDEF  NOBINOUT,CLCLOP,CO$BNO
001414                          CMDDEF  8BIT,STCLOP,CO$8BT
001422                          CMDDEF  EIGHT*BIT,STCLOP,CO$8BT
001430                          CMDDEF  NO8*BIT,CLCLOP,CO$8BT
001436                          CMDDEF  NOEIGHT*BIT,CLCLOP,CO$8BT
001444                          CMDDEF  7BIT,CLCLOP,CO$8BT
001452                          CMDDEF  PAR*ITY,STCLPD,CLPRHD
001460                          CMDDEF  NOPAR*ITY,STCLNP
001466                          CMDDEF  SEVEN*BIT,CLCLOP,CO$8BT
001474                          CMDDEF  TOP,CLTOP,0
001502                          CMDDEF  TR*ANSLATE,CLTRNS,0
001510                          CMDDEF  RES*ET,STCLRS,0
001516                          CMDDEF  SKI*P,STCLVL,CL$SKP
001524                          CMDDEF  WID*TH,STCLVL,CL$WID
001532                          CMDDEF  SP*EED,STCLSP,0
001540                          CMDDEF  VER*SION,STCLVR,0
001546                          CMDDEF  XON,STCLXN,0
001554                          TBLEND
                        ;
                        ;  Define options for SET CL PARITY= command
                        ;
000124                          TBLDEF  CLPR,2
001560                          CMDDEF  E*VEN,STCLPR,LP$PAR
001566                          CMDDEF  O*DD,STCLPR,LP$PAR!LP$ODD
001574                          CMDDEF  N*ONE,STCLPR,0
001602                          TBLEND
                        ;
                        ;  Define options for the SET HOST command
                        ;
000124                          TBLDEF  HOST,1
001606                          CMDDEF  DTE,HSTDTE
001612                          CMDDEF  CL,HSTDTE
001616                          CMDDEF  PO*RT,HSTPRT
001622                          TBLEND
                        ;
                        ;  Define options for SET PROCESS command
                        ;
000124                          TBLDEF  CPOP,2
001626                          CMDDEF  PRIV*ILEGES,PRVOPT,0
001634                          CMDDEF  ID*ENTIFICATION,CPPPID,0
001642                          CMDDEF  NAM*E,CPPNAM,0
001650                          CMDDEF  PRIO*RITY,CPPPRI,0
001656                          CMDDEF  SUSP*END,CPPFLG,CPFSUS,0
001664                          CMDDEF  RES*UME,CPPFLG,CPFRES,0
001672                          CMDDEF  AU*THORIZED,CPPFLG,CPFAUT,0
001700                          TBLEND
```

```
                                        .SBTTL  SET commands
                                        .SBTTL  .       Process
                                ;-----------------------------------------------------------------------
                                ;  Process the SET PROCESS command.
                                ;
000124                          SETPRC:
                                ;
                                ;  Initialize cells which will hold values parsed by the command
                                ;
000124  116767  000000G 177671          MOVB    CORUSR,CPPID    ;Set process ID=our job
000132  105067  177667                  CLRB    CPNAME          ;No name specified
000136  105067  177662                  CLRB    CPPRIO          ;No priority
000142  005067  177652                  CLR     CPFLAG          ;No flags
000146  004767  000000G                 CALL    CLRPRV          ;No privileges
                                ;
                                ;  Do command parsing
                                ;
000152  012704  001624'                 MOV     #CPOPHD,R4      ;Point to option driver list
000156  004767  000000G                 CALL    OPTLST          ;Parse the command
                                ;
                                ;  Now process each option that was accrued
                                ;
000162  004767  000366                  CALL    CPSNAM          ;See if process name was specified
000166  004767  000644                  CALL    CPSPRI          ;See if priority was specified
000172  004767  000464                  CALL    CPSPRV          ;See if privileges were specified
000176  004767  000676                  CALL    CPSFLG          ;See if flags were specified
                                ;
                                ;  Finished command
                                ;
000202  000167  000000G         9$:     JMP     RDCMD           ;Finished command
```

```
                                ;-------------------------------------------------------------------------
                                ;  SET PROCESS/IDENTIFICATION=job-number
                                ;
000206  010146                  CPPPID: MOV     R1,-(SP)
                                ;
                                ;  Accrue the ID value
                                ;
000210  004767  000000G                 CALL    ACRDEC
                                ;
                                ;  Make sure the ID value is valid
                                ;
000214  006301                          ASL     R1              ;Convert to word table index
000216  001413                          BEQ     10$             ;Invalid if zero
000220  020127  000000G                 CMP     R1,#LSTSL       ;Is this a valid job #?
000224  101010                          BHI     10$             ;Br if not
000226  032761  000000G 000000G         BIT     #$DILUP,LSW(R1) ;Is line logged on?
000234  001410                          BEQ     11$             ;Br if not
000236  110167  177561                  MOVB    R1,CPPID        ;Save ID value
                                ;
                                ;  Finished
                                ;
000242  012601                          MOV     (SP)+,R1
000244  000207                          RETURN
                                ;
                                ;  Invalid ID
                                ;
000246                          10$:    FABORT  #EM$ILN         ;Invalid job numver
000256                          11$:    FABORT  #NOTON          ;Line not logged on
```

```
                                ;--------------------------------------------------------------------------
                                ;  SET PROCESS/PRIORITY=value
                                ;
000266  010146                  CPPPRI: MOV     R1,-(SP)
000270  010546                          MOV     R5,-(SP)
                                ;
                                ;  Accrue the priority value
                                ;
000272  004767  000000G                 CALL    ACRDEC          ;Accrue priority value
                                ;
                                ;  See if the priority value is valid
                                ;
000276  005701                          TST     R1              ;Must be > 0
000300  003406                          BLE     10$             ;Br if invalid
000302  120167  000000G                 CMPB    R1,MXJPRI       ;Is it within valid range for job?
000306  101003                          BHI     10$             ;Br if not
000310  110167  177510                  MOVB    R1,CPPRIO       ;Save priority value
000314  000424                          BR      9$
                                ;
                                ;  Invalid priority value
                                ;
000316                          10$:    FERR    #BADPRI         ;Invalid priority value
000332  012705  000001                  MOV     #1,R5           ;Minimum prio value
000336  004767  000000G                 CALL    PRTDEC
000342                                  .PRINT  #TOTXT          ;"to"
000350  116705  000000G                 MOVB    MXJPRI,R5       ;Max possible
000354  004767  000000G                 CALL    PRTDEC
000360                                  .PRINT  #CRLF
                                ;
                                ;  Finished
                                ;
000366  012605                  9$:     MOV     (SP)+,R5
000370  012601                          MOV     (SP)+,R1
000372  000207                          RETURN
```

```
                                ;---------------------------------------------------------------------------
                                ;  SET PROCESS/NAME="string"
                                ;
000374  010246                  CPPNAM: MOV     R2,-(SP)
000376  010446                          MOV     R4,-(SP)
                                ;
                                ;  Make sure equal sign follows keyword
                                ;
000400  004767  000000G                 CALL    CHKEQ           ;Check for equal sign
000404  012702  000025'                 MOV     #CPNAME,R2      ;Point to buffer for name
                                ;
                                ;  See if string is enclosed in quote marks
                                ;
000410  004767  000000G                 CALL    SKPSPC          ;Skip over any spaces
000414  111300                          MOVB    (R3),R0         ;Get 1st char of string
000416  120027  000042                  CMPB    R0,#42          ;Quote mark?
000422  001403                          BEQ     3$              ;Br if yes
000424  120027  000047                  CMPB    R0,#47          ;Apostrophe?
000430  001014                          BNE     1$              ;Br if not
                                ;
                                ;  String is enclosed in quote marks
                                ;
000432  004767  000000G         3$:     CALL    ACRSTR          ;Accrue a quoted string
000436  005700                          TST     R0              ;Is string empty?
000440  001423                          BEQ     2$              ;If empty, go blank fill it.
000442  020027  000014                  CMP     R0,#12.         ;Is string too long?
000446  101026                          BHI     10$             ;Br if too long
000450  012704  000000G                 MOV     #BLKO,R4        ;Point to accrued string
000454  112422                  4$:     MOVB    (R4)+,(R2)+     ;Move to string buffer
000456  077002                          SOB     R0,4$
000460  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
                                ;
000462  111300                  1$:     MOVB    (R3),R0         ;Get next char from string
000464  001411                          BEQ     2$              ;Br if hit end of command
000466  120027  000057                  CMPB    R0,#'/          ;Start of next qualifier?
000472  001406                          BEQ     2$              ;Br if yes
000474  005203                          INC     R3              ;Point to next character in command
000476  020227  000041'                 CMP     R2,#CPNAME+12.  ;Buffer overflow?
000502  103010                          BHIS    10$             ;Br if yes
000504  110022                          MOVB    R0,(R2)+        ;Store char into name buffer
000506  000765                          BR      1$
                                ;
                                ;  Fill remainder of buffer with blanks
                                ;
000510  020227  000041'         2$:     CMP     R2,#CPNAME+12.  ;Filled buffer yet?
000514  103011                          BHIS    9$              ;Br if yes
000516  112722  000040                  MOVB    #' ,(R2)+       ;Pad with blanks
000522  000772                          BR      2$
                                ;
                                ;  Error: string is too long
                                ;
000524                          10$:    FERR    #EM$STL         ;String is too long
                                ;
                                ;  Finished
```

```
                                   ;
000540  012604                     9$:     MOV     (SP)+,R4
000542  012602                             MOV     (SP)+,R2
000544  000207                             RETURN
                                   ;-------------------------------------------------------------------------
                                   ;  Set some flag for a SET PROCESS qualifier
                                   ;
000546  051467  177246             CPPFLG: BIS     (R4),CPFLAG     ;Set specified flag bit
000552  000207                             RETURN
```

```
                                       ;---------------------------------------------------------------------------
                                       ;  See if NAME qualifier was specified.
                                       ;
000554  010146                         CPSNAM: MOV     R1,-(SP)
000556  010246                                 MOV     R2,-(SP)
                                       ;
                                       ;  See if name qualifier was specified
                                       ;
000560  105767  177241                         TSTB    CPNAME          ;Was name qualifier specified?
000564  001423                                 BEQ     9$              ;Br if not
                                       ;
                                       ;  Make sure we are changing for our own process
                                       ;
000566  116701  177231                         MOVB    CPPID,R1        ;Get job index
000572  120167  000000G                        CMPB    R1,CORUSR       ;Affecting our own process?
000576  001021                                 BNE     10$             ;Br if not
000600  032767  000000G 000000G                BIT     #PO$NAM,PRIVCO  ;Are we authorized to change our name?
000606  001421                                 BEQ     11$             ;Br if not
000610  070127  000006                         MUL     #6.,R1          ;Each job has 12 character name
000614  062701  000000G                        ADD     #LUNAME,R1      ;Point to name cell for our job
000620  012702  000025'                        MOV     #CPNAME,R2      ;Point to name buffer
000624  012700  000014                         MOV     #12.,R0         ;Get # bytes to move
000630  112221                         1$:     MOVB    (R2)+,(R1)+     ;Set name for job
000632  077002                                 SOB     R0,1$
                                       ;
                                       ;  Finished
                                       ;
000634  012602                         9$:     MOV     (SP)+,R2
000636  012601                                 MOV     (SP)+,R1
000640  000207                                 RETURN
                                       ;
                                       ;  Cannot change name of another job
                                       ;
000642                                 10$:    FABORT  #EM$CNO
000652                                 11$:    FABORT  #EM$NPR
```

```
                                 ;----------------------------------------------------------------------
                                 ;  Change privilege flags for our process.
                                 ;
000662  010246                   CPSPRV: MOV     R2,-(SP)
000664  010346                           MOV     R3,-(SP)
000666  010446                           MOV     R4,-(SP)
                                 ;
                                 ;  See if any privilege changes were specified
                                 ;
000670  012702  000000G                  MOV     #PFSO,R2        ;Flags to set
000674  012703  000000G                  MOV     #PFCO,R3        ;Flags to clear
000700  012700  000000G                  MOV     #PVNPW,R0       ;# words to check
000704  005722                   1$:     TST     (R2)+           ;Any flags to set?
000706  001004                           BNE     2$              ;Br if yes
000710  005723                           TST     (R3)+           ;Any flags to clear?
000712  001002                           BNE     2$              ;Br if yes
000714  077005                           SOB     R0,1$           ;Keep checking
000716  000433                           BR      9$              ;No privilege changes requested
                                 ;
                                 ;  Make sure we are changing privileges for our job
                                 ;
000720  126767  177077  000000G  2$:     CMPB    CPPID,CORUSR    ;Changing privilege for our job?
000726  001033                           BNE     10$             ;Br if not
                                 ;
                                 ;  Set correct EMT function code depending on whether we are doing a
                                 ;  normal privilege change or are changing authorized privileges.
                                 ;
000730  012704  000001                   MOV     #1,R4           ;Assume normal priv change
000734  032767  000004  177056           BIT     #CPFAUT,CPFLAG  ;Changing authorized privileges?
000742  001402                           BEQ     4$              ;Br if not
000744  012704  000002                   MOV     #2,R4           ;Code to change authorized privileges
000750  110467  177123           4$:     MOVB    R4,PVSEMT+3     ;Set code in EMT arg block
000754  110467  177127                   MOVB    R4,PVCEMT+3
                                 ;
                                 ;  Set any specified privileges
                                 ;
000760  005004                           CLR     R4              ;No privilege error detected
000762  012700  000074'                  MOV     #PVSEMT,R0      ;Point to EMT arg block
000766  104375                           EMT     375             ;Set privilege flags
000770  103001                           BCC     3$              ;Br if no authorization violation
000772  005204                           INC     R4              ;Remember we had authorization violation
                                 ;
                                 ;  Clear any specified privileges
                                 ;
000774  012700  000104'          3$:     MOV     #PVCEMT,R0      ;Point to EMT arg block
001000  104375                           EMT     375             ;Clear privilege flags
001002  005704                           TST     R4              ;Did we have an authorization violation?
001004  001010                           BNE     11$             ;Br if yes
                                 ;
                                 ;  Finished
                                 ;
001006  012604                   9$:     MOV     (SP)+,R4
001010  012603                           MOV     (SP)+,R3
001012  012602                           MOV     (SP)+,R2
001014  000207                           RETURN
                                 ;
                                 ;  Can't change privilege for another job
```

```
                              ;
   001016                     10$:    FABORT  #EM$CPO         ;Can't change privilege for another job
                              ;
                              ;  Not authorized to give yourself this privilege
                              ;
   001026                     11$:    FABORT  #EM$CAP         ;Can't authorize that privilege
```

```
                                        ;-----------------------------------------------------------------------
                                        ;  See if we need to change priority for job.
                                        ;
 001036  010246                         CPSPRI: MOV     R2,-(SP)
                                        ;
                                        ;  See if a priority change was requested
                                        ;
 001040  105767  176760                         TSTB    CPPRIO          ;Was a priority specified
 001044  001413                                 BEQ     9$              ;Br if not
 001046  116702  176751                         MOVB    CPPID,R2        ;Get ID # of job we want to affect
 001052  004767  000000G                        CALL    CKACOJ          ;Can we access that job?
 001056  103406                                 BCS     9$              ;Br if not
                                        ;
                                        ;  Change priority value
                                        ;
 001060  116762  176740  000000G                MOVB    CPPRIO,LBSPRI(R2) ;Set base priority
 001066  116762  176732  000000G                MOVB    CPPRIO,LPRI(R2) ;Set running priority
                                        ;
                                        ;  Finished
                                        ;
 001074  012602                         9$:     MOV     (SP)+,R2
 001076  000207                                 RETURN
```

```
                                        ;-------------------------------------------------------------------------
                                        ;  See if we need to suspend or resume a job.
                                        ;
 001100  010246                         CPSFLG: MOV     R2,-(SP)
                                        ;
                                        ;  See if any flags were specified
                                        ;
 001102  032767  000003  176710                 BIT     #<CPFSUS!CPFRES>,CPFLAG ;Suspend or resume requested?
 001110  001423                                 BEQ     9$              ;Br if not
                                        ;
                                        ;  Make sure we aren't suspending or resuming our own job
                                        ;
 001112  116702  176705                         MOVB    CPPID,R2        ;Get ID # of job we want to affect
 001116  120267  000000G                        CMPB    R2,CORUSR       ;Is it our job?
 001122  001420                                 BEQ     10$             ;Br if yes
                                        ;
                                        ;  Make sure we can access the specified job
                                        ;
 001124  004767  000000G                        CALL    CKACOJ          ;Can we access that job?
 001130  103413                                 BCS     9$              ;Br if not
                                        ;
                                        ;  Suspend or resume the specified job
                                        ;
 001132  012700  000000G                        MOV     #RJEMT,R0       ;Point to resume-job arg block
 001136  032767  000002  176654                 BIT     #CPFRES,CPFLAG  ;Does he want to resume the job?
 001144  001002                                 BNE     1$              ;Br if yes
 001146  012700  000000G                        MOV     #SJEMT,R0       ;Point to suspend-job arg block
 001152  010260  000004                 1$:     MOV     R2,4(R0)        ;Set job # in EMT arg block
 001156  104375                                 EMT     375             ;Suspend or resume the job
                                        ;
                                        ;  Finished
                                        ;
 001160  012602                         9$:     MOV     (SP)+,R2
 001162  000207                                 RETURN
                                        ;
                                        ;  Error -- Trying to suspend or resume our own job
                                        ;
 001164                                 10$:    FABORT  #EM$IDR         ;/ID parameter required
```

```
                                        .SBTTL  .       Terminal
                                ;-----------------------------------------------------------------
                                ;  Process the SET TERMINAL command.
                                ;
001174                          SETTTY:
                                ;
                                ;  See if a line number was specified
                                ;
001174  105067  176622                  CLRB    SETPRM          ;Assume no line number specified
001200  004767  000000G                 CALL    SKPSPC          ;Skip over any spaces
001204  010305                          MOV     R3,R5           ;Save pointer to first operand
001206  005002                          CLR     R2              ;Say no digits seen yet
001210  112500                  1$:     MOVB    (R5)+,RO        ;Get next char
001212  001413                          BEQ     2$              ;Br if hit end of command line
001214  120027  000040                  CMPB    RO,#40          ;Is this a space?
001220  001410                          BEQ     2$              ;Br if yes
001222  120027  000060                  CMPB    RO,#'0          ;Is this a digit?
001226  103435                          BLO     3$              ;Br if not
001230  120027  000071                  CMPB    RO,#'9
001234  101032                          BHI     3$              ;Br if not digit
001236  005202                          INC     R2              ;Remember we saw a digit
001240  000763                          BR      1$              ;Scan first item and see if it is a number
001242  005702                  2$:     TST     R2              ;Was first item a number?
001244  001426                          BEQ     3$              ;Br if not
                                ;
                                ;  There is a terminal number specified, accrue it
                                ;
001246  004767  000000G                 CALL    ACRDEC          ;Accrue the terminal number
                                ;
                                ;  See if this is a valid terminal number
                                ;
001252  006301                          ASL     R1              ;Convert line number to line index number
001254  001403                          BEQ     4$              ;Zero is not valid line number
001256  020127  000000G                 CMP     R1,#LSTPL       ;Is this a primary line number?
001262  101404                          BLOS    5$              ;Br if yes
001264                          4$:     FABORT  #EM$ILN         ;Invalid line number
                                ;
                                ;  Require TERMINAL privilege to make any permanent change
                                ;
001274  004767  000000G         5$:     CALL    CKTERM          ;Require TERMINAL privilege to do this
001300  110167  176516                  MOVB    R1,SETPRM       ;Remember to make set permanent
                                ;
                                ;  If we are changing line parameters for a line that is not
                                ;  currently logged on, initialize the LSW2 table from the initial
                                ;  flags for the line (ILSW2).
                                ;
001304  032761  000000G 000000G         BIT     #$DILUP,LSW(R1) ;Is line logged on?
001312  001003                          BNE     3$              ;Br if yes
001314  016161  000000G 000000G         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.
                                ;
001322  012704  000000'         3$:     MOV     #TTHD,R4        ;Get pointer to option table
001326  004767  000000G                 CALL    SCNOPS          ;Process all of the command options
                                ;
```

```
     58                                  ;   End of command reached
     59                                  ;
     60 001332  000167  000000G                  JMP     RDCMD           ;Finished SET TT command
```

```
                                         ;-------------------------------------------------------------------
                                         ;  SET TERMINAL n START
                                         ;
001336  004767  000000G                  STTTST: CALL    CKPRIV           ;Require OPER privilege
                                         ;
                                         ;  See if line is already started
                                         ;
001342  032761  000000G 000000G                  BIT     #$DILUP,LSW(R1)  ;Is line already started?
001350  001404                                   BEQ     1$               ;Br if not
001352                                           FABORT  #EM$LAS          ;Line is already started
                                         ;
                                         ;  See if line is marked dead
                                         ;
001362  032761  000000G 000000G          1$:     BIT     #$DEAD,LSW3(R1)  ;Is line marked as dead?
001370  001404                                   BEQ     2$               ;Br if not
001372                                           FABORT  #EM$LFD          ;Line is dead
                                         ;
                                         ;  See if a CL unit is attached to the line
                                         ;
001402  005761  000000G                  2$:     TST     LCLUNT(R1)       ;Is line being used by a CL unit?
001406  002404                                   BLT     3$               ;Br if not
001410                                           FABORT  #EM$CLU          ;Line in use by a CL unit
                                         ;
                                         ;  Start the line
                                         ;
001420  010167  176474                   3$:     MOV     R1,STAEMT+4      ;Store line index into EMT arg block
001424  012700  000114'                          MOV     #STAEMT,R0       ;Point to EMT arg block
001430  104375                                   EMT     375              ;Start the line
001432  000207                           9$:     RETURN
```

```
                                          ;---------------------------------------------------------------------
                                          ;   SET TERMINAL [n] SPEED=n
                                          ;
001434  004767  000000G                   STTTSP: CALL    CKTERM              ;Require TERMINAL privilege
001440  004767  000000G                           CALL    ACRSPD              ;Accrue speed value
001444  042761  000000G 000000G                   BIC     #$AUTO,LSW2(R1)     ;Say no autobaud on this line
001452  116100  000001G                           MOVB    LMXPRM+1(R1),R0     ;Get control flags for line
001456  042700  000000G                           BIC     #<LP$SPD>,R0        ;Clear speed flags
001462  050005                                    BIS     R0,R5               ;Combine parity and length flags with speed
001464  010567  176372                    STSP1:  MOV     R5,LSPEMT+4         ;Store speed, par, len code into EMT arg block
001470  010167  176364                            MOV     R1,LSPEMT+2         ;Store line # into EMT arg block
001474  006267  176360                            ASR     LSPEMT+2            ;Convert index # to line #
001500  012700  000056'                           MOV     #LSPEMT,R0          ;Point to EMT arg block
001504  104375                                    EMT     375                 ;Set the speed
001506  103402                                    BCS     2$                  ;Br if error
001510  000167  000474                            JMP     BS01
                                          ;
                                          ;   Error on EMT to set speed
                                          ;
001514                                    2$:     FABORT  #EM$ILN             ;Say invalid line number
```

```
                                       ;------------------------------------------------------------------
                                       ;   SET TT [n] XON
                                       ;
  001524  004767  000000G              SETXON: CALL     CKTERM           ;Require TERMINAL privilege
  001530  010167  176332                       MOV      R1,XONEMT+2      ;Set line # in EMT argument block
  001534  006267  176326                       ASR      XONEMT+2         ;Convert index # to line #
  001540  012700  000064'                      MOV      #XONEMT,R0       ;Point to EMT argument block
  001544  104375                               EMT      375              ;Do the XON EMT
  001546  103402                               BCS      2$               ;Br if invalid line #
  001550  000167  000434                       JMP      BS01
  001554                               2$:     FABORT   #EM$ILN          ;Say invalid line number
```

```
                                        ;-----------------------------------------------------------------
                                        ;  SET TT [n] [NO]DTR
                                        ;
 001564  112767  000002  176276  SETDTR: MOVB    #2,DTREMT          ;Set function to raise DTR
 001572  000403                          BR      DTRCOM             ;Br to common routine
 001574  112767  000003  176266  CLRDTR: MOVB    #3,DTREMT          ;Set function to lower DTR
 001602  004767  000000G         DTRCOM: CALL    CKTERM             ;Require TERMINAL privilege
 001606  006201                          ASR     R1                 ;Convert line index to line number
 001610  010167  176256                  MOV     R1,DTREMT+2        ;Set in EMT  arg block
 001614  006301                          ASL     R1                 ;Get back as line index
 001616  012700  000070'                 MOV     #DTREMT,R0         ;Point to EMT arg block to
 001622  104375                          EMT     375                ;Raise or lower DTR
 001624  103402                          BCS     2$                 ;Br on EMT error
 001626  000167  000356                  JMP     BS01
 001632                          2$:     FABORT  EM$ILN             ;Say invalid line number
```

```
                                              ;-------------------------------------------------------------
                                              ;   SET TERMINAL AUTODAUD
                                              ;
001642  004767  000000G                       SAUTO:   CALL     CKTERM              ;Require TERMINAL privilege
001646  052761  000000G 000000G                        BIS      #$AUTO,LSW2(R1) ;Set flag saying to do autobaude select
001654  032761  000000G 000000G                        BIT      #$DILUP,LSW(R1) ;Is anyone using the line now?
001662  001003                                         BNE      9$                  ;Br if yes
001664  012705  000000G                                MOV      #S9600,R5           ;Set line speed to 9600 baud
001670  000675                                         BR       STSP1
001672  000167  000312                        9$:      JMP      BS01
```

```
                                   ;-----------------------------------------------------------------------
                                   ;  SET TERMINAL PARITY={EVEN ODD NONE}
                                   ;
001676  010446                     SETPAR: MOV     R4,-(SP)        ;Save current option table pointer
001700  004767  000000G                    CALL    CKTERM          ;Require TERMINAL privilege
001704  004767  000000G                    CALL    SKPSPC          ;Skip past any spaces
001710  121327  000075                     CMPB    (R3),#'=        ;Equal sign following PARITY?
001714  001001                             BNE     1$              ;Br if not
001716  005203                             INC     R3              ;Skip past equal sign
001720  012704  001064'            1$:     MOV     #PARHD,R4       ;Point to new option table
001724  004767  000000G                    CALL    SETWRD          ;Process EVEN, ODD, NONE word
001730  012604                             MOV     (SP)+,R4
001732  000207                             RETURN
```

```
                                      ;-----------------------------------------------------------------------
                                      ;   SET TERMINAL BITS=n
                                      ;
 001734  004767  000000G              STCLEN: CALL     CKTERM              ;Require TERMINAL privilege
 001740  116105  000001G                      MOVB     LMXPRM+1(R1),R5     ;Get current line parameters
 001744  010102                               MOV      R1,R2               ;Save line index number
 001746  004767  000000G                      CALL     ACRDEC              ;Accrue parameter value
 001752  042705  000000G                      BIC      #LP$7BT,R5          ;Assume 8 bits wanted
 001756  020127  000007                       CMP      R1,#7.              ;Is value 7?
 001762  001003                               BNE      1$                  ;Br if not
 001764  052705  000000G                      BIS      #LP$7BT,R5          ;Select 7 bits
 001770  000407                               BR       2$
 001772  120127  000010               1$:     CMPB     R1,#8.              ;Is value 8?
 001776  001404                               BEQ      2$                  ;Br if yes
 002000                                       FABORT   #EM$ICL             ;Invalid character length
 002010  010201                       2$:     MOV      R2,R1               ;Get back line index number
 002012  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.
                                     ;
002016  004767  000000G              STPAR:  CALL    CKTERM              ;Require TERMINAL privilege
002022  116105  000001G                      MOVB    LMXPRM+1(R1),R5 ;Get current flags for line
002026  042705  000000C                      BIC     #LP$PAR!LP$ODD,R5;Clear parity control flags
002032  151405                               BISB    @R4,R5              ;Set new flags
002034  000167  177424                       JMP     STSP1               ;Go set value
                                     ;
                                     ;  SET TERMINAL NOPARITY
                                     ;
002040  004767  000000G              STNOPR: CALL    CKTERM              ;Require TERMINAL privilege
002044  116105  000001G                      MOVB    LMXPRM+1(R1),R5 ;Get current flags for line
002050  042705  000000C                      BIC     #LP$PAR!LP$ODD,R5;Clear parity control flags
002054  000167  177404                       JMP     STSP1               ;Go set value

                                     ;----------------------------------------------------------------
                                     ;  SET TERMINAL [n] TRANSLATE=(ext=int,ext=int,...)
                                     ;
002060  010146                       STTRNS: MOV     R1,-(SP)            ;Save original line number
002062  016101  000000G                      MOV     LNPRIM(R1),R1       ;Make sure we have primary line number
002066  004767  002156                       CALL    PRSTRN              ;Parse the translate qualifier
002072  012601                               MOV     (SP)+,R1            ;Restore original line number
002074  000207                               RETURN
```

```
                                                ;------------------------------------------------------------
                                                ;   Set parameters for particular terminal types.
                                                ;
                                                ;   SET PARAMETERS FOR A VT50/VT52.
                                                ;   SET VT50
002076  042761  000000G 000000G SVT50:  BIC     #VT52NO,LSW2(R1);CLEAR SOME FLAGS
002104  052761  000000G 000000G         BIS     #VT52FL,LSW2(R1);SAY TERMINAL IS A VT50
002112  012761  000000G 000000G         MOV     #VT52,LTRMTP(R1);SET TERMINAL TYPE
002120  000414                          BR      BS02
                                                ;
                                                ;   SET PARAMETERS FOR LA36.
                                                ;
002122  042761  000000G 000000G STLA36: BIC     #LA36NO,LSW2(R1);RESET VARIOUS FLAGS
002130  052761  000000G 000000G         BIS     #LA36FL,LSW2(R1)
002136  012761  000000G 000000G         MOV     #LA36,LTRMTP(R1);SET TERMINAL TYPE
002144  042761  000000G 000000G         BIC     #$SLON,LSW7(R1) ;Disalble SL editor for this terminal type
                                                ;
                                                ;   Set up terminal type information
                                                ;
002152  120167  000000G                 BS02:   CMPB    R1,CORUSR       ;Changing terminal type for our job?
002156  001006                                  BNE     1$              ;Br if not our job
002160  005761  000000G                         TST     LWINDO(R1)      ;Is process windowing turned on for job?
002164  001403                                  BEQ     1$              ;Br if not
002166  012700  000122'                         MOV     #WINSTT,R0      ;Point to EMT argument block
002172  104375                                  EMT     375             ;Tell windowing system about terminal type
002174  105767  175622                  1$:     TSTB    SETPRM          ;Should we make the set permanent?
002200  001403                                  BEQ     BS01            ;Br if not
002202  016161  000000G 000000G                 MOV     LTRMTP(R1),ITRMTP(R1) ;Set initial terminal type
002210  016161  000000G 000000G BS01:   MOV     LSW2(R1),LSW2S(R1) ;Save in case of ctrl-c reentry
002216  105767  175600                          TSTB    SETPRM          ;Should we make the set permanent?
002222  001403                                  BEQ     SETJMP          ;Br if not
002224  016161  000000G 000000G                 MOV     LSW2(R1),ILSW2(R1);Set initial flags for line
002232  000207                          SETJMP: RETURN                  ;Return to process next qualifier
                                                ;
                                                ;   LA120 parameters
                                                ;
002234  042761  000000G 000000G SLA120: BIC     #LA12NO,LSW2(R1);SET FLAGS
002242  052761  000000G 000000G         BIS     #LA12FL,LSW2(R1)
002250  012761  000000G 000000G         MOV     #LA120,LTRMTP(R1);SET TERMINAL TYPE
002256  042761  000000G 000000G         BIC     #$SLON,LSW7(R1) ;Disalble SL editor for this terminal type
002264  000732                          BR      BS02
                                                ;
                                                ;   VT100 parameters
                                                ;
002266  042761  000000G 000000G SVT100: BIC     #VT10NO,LSW2(R1);SET FLAGS
002274  052761  000000G 000000G         BIS     #VT10FL,LSW2(R1)
002302  012761  000000G 000000G         MOV     #VT100,LTRMTP(R1);SET TERMINAL TYPE
002310  000720                  STERM:  BR      BS02
                                                ;
                                                ;   VT200 parameters
                                                ;
002312  042761  000000G 000000G SVT200: BIC     #VT20NO,LSW2(R1)
002320  052761  000000G 000000G         BIS     #VT20FL,LSW2(R1)
002326  012761  000000G 000000G         MOV     #VT200,LTRMTP(R1)
002334  000706                          BR      BS02
                                                ;
                                                ;   Hazeltine parameters
```

```
                                           ;
002336  042761  000000G 000000G STHAZL: BIC     #HAZLNO,LSW2(R1);RESET SOME FLAGS
002344  052761  000000G 000000G         BIS     #HAZLFL,LSW2(R1);SET SOME FLAGS
002352  012761  000000G 000000G         MOV     #HAZEL,LTRMTP(R1);SET TERMINAL TYPE
002360  042761  000000G 000000G         BIC     #$SLON,LSW7(R1) ;Disalble SL editor for this terminal type
002366  000671                          BR      BS02
                                ;
                                ;   ADM3A parameters
                                ;
002370  042761  000000G 000000G SADM3A: BIC     #ADM3NO,LSW2(R1);SET FLAGS
002376  052761  000000G 000000G         BIS     #ADM3FL,LSW2(R1)
002404  012761  000000G 000000G         MOV     #ADM3A,LTRMTP(R1);SET TERMINAL TYPE
002412  042761  000000G 000000G         BIC     #$SLON,LSW7(R1) ;Disalble SL editor for this terminal type
002420  000654                          BR      BS02
                                ;
                                ;   SET PARAMETERS FOR DIABLO-1620 TERMINAL
                                ;
002422                          SQUME:
002422  042761  000000G 000000G SDIAB:  BIC     #DIABNO,LSW2(R1);RESET VARIOUS FLAGS
002430  052761  000000G 000000G         BIS     #DIABFL,LSW2(R1);SAY THIS IS A DIABLO
002436  012761  000000G 000000G         MOV     #DIABLO,LTRMTP(R1);SET TERMINAL TYPE
002444  042761  000000G 000000G         BIC     #$SLON,LSW7(R1) ;Disalble SL editor for this terminal type
002452  000637                          BR      BS02
                                ;
                                ;   SET TT QUIET
                                ;   (Also, SET NOVERIFY)
                                ;
002454  116701  000000G         STNOVR: MOVB    CORUSR,R1               ;Get current job index number
002460  004767  000004                  CALL    SETQUT                  ;Do the set
002464  000167  000000G                 JMP     RDCMD                   ;Finished
                                ;
002470  052761  000000G 000000G SETQUT: BIS     #$QTSET,LSW2(R1)        ;REMEMBER SET WAS DONE
002476  052761  000000G 000000G         BIS     #$QUIET,LSW4(R1)        ;HAVE IMMEDIATE EFFECT
002504  120167  000000G                 CMPB    R1,CORUSR               ;Are we doing set for current job?
002510  001025                          BNE     1$                      ;Br if not
002512                                  .GVAL   #XAREA,#<CFSTS-MONVEC>  ;GET CURRENT COMMAND FILE FLAGS
002532  010002                          MOV     R0,R2
002534  052702  000000G                 BIS     #CF$QUT,R2              ;SET TT-QUIET FLAG
002540                                  .PVAL   #XAREA,#<CFSTS-MONVEC>,R2;STORE UPDATED FLAGS
002564  000167  177420          1$:     JMP     BS01
                                ;
                                ;   SET TT NOQUIET
                                ;   (Also, SET VERIFY)
                                ;
002570  116701  000000G         STVRFY: MOVB    CORUSR,R1               ;Get current job index number
002574  004767  000004                  CALL    RSTQUT                  ;Do the SET
002600  000167  000000G                 JMP     RDCMD                   ;Finished
                                ;
002604  042761  000000G 000000G RSTQUT: BIC     #$QTSET,LSW2(R1)        ;REMEMBER SET WAS DONE
002612  042761  000000G 000000G         BIC     #$QUIET,LSW4(R1)        ;START LISTING NOW
002620  120167  000000G                 CMPB    R1,CORUSR               ;Are we doing set for current line?
002624  001025                          BNE     1$                      ;Br if not
002626                                  .GVAL   #XAREA,#<CFSTS-MONVEC>  ;GET CURRENT COMMAND FILE FLAGS
002646  010002                          MOV     R0,R2
002650  042702  000000G                 BIC     #CF$QUT,R2              ;CLEAR TT-QUIET FLAG
002654                                  .PVAL   #XAREA,#<CFSTS-MONVEC>,R2;STORE UPDATED FLAGS
002700  000167  177304          1$:     JMP     BS01
```

```
                                               ;
                                               ;  SET TT NOSCOPE
                                               ;
002704  042761  000000G 000000G SETNSC: BIC     #$SCOPE,LSW2(R1)        ;Say this is not a scope type terminal
002712  042761  000000G 000000G         BIC     #$SLON,LSW7(R1)         ;Disable SL editor
002720  000167  177264                  JMP     BS01
                                               ;
                                               ;  SET TT WAIT
                                               ;
002724  042761  000000G 000000G SETTW:  BIC     #$SNWTT,LSW6(R1);UNDO SET TT NOWAIT
002732  042761  000000G 000000G         BIC     #$NOWTT,LSW5(R1)
002740  000207                          RETURN
                                               ;
                                               ;  SET TT NOWAIT
                                               ;
002742  052761  000000G 000000G SETTNW: BIS     #$SNWTT,LSW6(R1);SET TT NOWAIT
002750  052761  000000G 000000G         BIS     #$NOWTT,LSW5(R1)
002756  000207                          RETURN
                                               ;
                                               ;  SET TT DEAD
                                               ;
002760  004767  000000G         SETDED: CALL    CKTERM          ;Require TERMINAL privilege
002764  032761  000000G 000000G         BIT     #$DILUP,LSW(R1) ;Is line active now
002772  001406                          BEQ     SETTTB          ;Br if not
002774                                  FABORT  #RUNMS          ;Can't kill an active line
                                               ;
                                               ;  Perform terminal sets that only involve setting or clearing flag bits.
                                               ;
003004  004767  000000G         SETTTP: CALL    CKTERM          ;Require TERMINAL privilege
003010  012405                  SETTTB: MOV     (R4)+,R5        ;POINT TO TABLE
003012  060105                          ADD     R1,R5           ;POINT TO TABLE ENTRY FOR THIS USER
003014  051415                          BIS     @R4,@R5         ;SET THE DESIRED FLAG
003016  000167  177166                  JMP     BS01
                                               ;
                                               ;  RESET A BIT IN A USER TABLE
                                               ;
003022  004767  000000G         CLRTTP: CALL    CKTERM          ;Require TERMINAL privilege
003026  012405                  CLRTTB: MOV     (R4)+,R5        ;POINT TO TABLE
003030  060105                          ADD     R1,R5           ;POINT TO TABLE ENTRY FOR THIS USER
003032  041415                          BIC     @R4,@R5         ;RESET THE DESIRED FLAG
003034  000167  177150                  JMP     BS01
```

```
                                        .SBTTL  .       CL
                               ;-----------------------------------------------------------------
                               ;  Process the SET CLn command
                               ;
                               ;  Inputs:
                               ;   R2 = CL unit number
                               ;
003040  004767  000000G        SETCL:  CALL    CKTERM          ;Require TERMINAL privilege for SET CL
003044  020227  000000G                CMP     R2,#CLTOTL      ;Is this a valid unit?
003050  103404                         BLO     3$              ;Br if ok
003052                                 FABORT  #EM$IUN         ;Invalid CL unit
003062  006302                 3$:     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.
                               ;
003064  004767  000000G        1$:     CALL    SKPSPC          ;Skip over any spaces
003070  111300                         MOVB    (R3),R0         ;Get next character from command
003072  001434                         BEQ     15$             ;Br if hit the end of the command
003074  120027  000054                 CMPB    R0,#',          ;Comma separator?
003100  001403                         BEQ     4$              ;Br if yes
003102  120027  000057                 CMPB    R0,#'/          ;Slash separator?
003106  001005                         BNE     5$              ;Br if not
003110  005203                 4$:     INC     R3              ;Skip past separator
003112  004767  000000G                CALL    SKPSPC          ;Skip over any spaces
003116  105713                         TSTB    (R3)            ;Are we at the end of the command now?
003120  001415                         BEQ     10$             ;Br if yes -- Error
                               ;
                               ;  Process the next command qualifier
                               ;
003122  012704  001120'        5$:     MOV     #CLOPHD,R4      ;Point to option table
003126  004767  000000G                CALL    SEARCH          ;Search for option keyword
003132  103404                         BCS     11$             ;Br if cannot idenify keyword
003134  010246                         MOV     R2,-(SP)        ;Save unit index
003136  004734                         CALL    @(R4)+          ;Call routine to process qualifier
003140  012602                         MOV     (SP)+,R2        ;Get back unit number index
003142  000750                         BR      1$              ;Go see if there are more qualifiers
                               ;
                               ;  Invalid qualifier keyword
                               ;
003144                         11$:    FABORT  #CSIMS1         ;Invalid option keyword
                               ;
                               ;  Invalid command syntax
                               ;
003154                         10$:    FABORT  #EM$CSE         ;Command syntax error
                               ;
                               ;  End of command reached
                               ;
003164  000167  000000G        15$:    JMP     RDCMD           ;Finished command
                               ;
                               ;  CL unit is not assigned to a line
                               ;
003170                         STCLNL: FABORT  #EM$CLN         ;CL unit is not assigned to a line
                               ;
                               ;  Turn on some option
                               ;
003200  005762  000000G        STCLOP: TST     CL$LIX(R2)      ;Is this CL unit assigned to a line?
```

```
     58 003204  001771                      BEQ     STCLNL           ;Br if not
     59 003206  051462  000000G             BIS     (R4),CL$OPT(R2)  ;Set desired option flag
     60 003212  000207                      RETURN
     61                             ;
     62                             ;  Turn off some option
     63                             ;
     64 003214  005762  000000G     CLCLOP: TST     CL$LIX(R2)       ;Is this CL unit assigned to a line?
     65 003220  001763                      BEQ     STCLNL           ;Br if not
     66 003222  041462  000000G             BIC     (R4),CL$OPT(R2)  ;Reset the option
     67 003226  000207                      RETURN
     68                             ;
     69                             ;  Set a parameter value
     70                             ;
     71 003230  005762  000000G     STCLVL: TST     CL$LIX(R2)       ;Is this CL unit assigned to a line?
     72 003234  001755                      BEQ     STCLNL           ;Br if not
     73 003236  004767  000000G             CALL    ACRDEC           ;Accrue decimal value
     74 003242  011404                      MOV     (R4),R4          ;Get address of table to store value into
     75 003244  060204                      ADD     R2,R4            ;Point to correct table entry
     76 003246  010114                      MOV     R1,(R4)          ;Store value into table
     77 003250  000207                      RETURN
     78                             ;
     79                             ;  SET CL [NO]GRAPH
     80                             ;
     81 003252  005762  000000G     STCLGR: TST     CL$LIX(R2)       ;Is this CL unit assigned to a line?
     82 003256  001744                      BEQ     STCLNL           ;Br if not
     83 003260  011462  000000G             MOV     (R4),CL$WID(R2)  ;Set width to 0 (graph) or 132 (nograph)
     84 003264  000207                      RETURN
     85                             ;
     86                             ;  SET CL TOP
     87                             ;
     88 003266  005762  000000G     CLTOP:  TST     CL$LIX(R2)       ;Is this CL unit assigned to a line?
     89 003272  001736                      BEQ     STCLNL           ;Br if not
     90 003274  005062  000000G             CLR     CL$LIN(R2)       ;Say we are at top of the page
     91 003300  000207                      RETURN
     92                             ;
     93                             ;  SET CL SPEED=value
     94                             ;
     95 003302  016202  000000G     STCLSP: MOV     CL$LIX(R2),R2    ;Is this CL unit assigned to a line?
     96 003306  001730                      BEQ     STCLNL           ;Br if not
     97 003310  004767  000000G             CALL    ACRSPD           ;Accrue speed value
     98 003314  116200  000001G             MOVB    LMXPRM+1(R2),R0  ;Get current flags for line
     99 003320  042700  000000G             BIC     #LP$SPD,R0       ;Clear speed code
    100 003324  050005                      BIS     R0,R5            ;Put in parity and length flags
    101 003326  010567  174530      STCLS1: MOV     R5,LSPEMT+4      ;Store speed code into EMT arg block
    102 003332  006202                      ASR     R2               ;Convert line index # to line #
    103 003334  010267  174520              MOV     R2,LSPEMT+2      ;Store line # into EMT arg block
    104 003340  012700  000056'             MOV     #LSPEMT,R0       ;Point to EMT arg block
    105 003344  104375                      EMT     375              ;Set the speed
    106 003346  103403                      BCS     2$               ;Br if error
    107 003350  000207                      RETURN
    108                             ;
    109                             ;  Error on EMT to set speed
    110                             ;
    111 003352                      2$:     FABORT  #EM$IUN          ;Say invalid unit number
    112                             ;
    113                             ;  SET CLn XON
    114                             ;
    003362  012705  000000G         STCLXN: MOV     #CLSFCH,R5      ;Get .SPFUN code
    003366  000402                          BR      CLSPFN          ;Do the .SPFUN
                                    ;
                                    ;   SET CLn RESET
                                    ;
    003370  012705  000000G         STCLRS: MOV     #CLSFRS,R5      ;Get .SPFUN code
                                    ;
                                    ;   Do a .SPFUN to a CL unit.
                                    ;   R5 = .SPFUN code.
                                    ;
    003374  005762  000000G         CLSPFN: TST     CL$LIX(R2)      ;Is this CL unit assigned to a line?
    003400  001673                          BEQ     STCLNL          ;Br if not
    003402  004767  001040                  CALL    CLLOOK          ;Open channel 1 to CL unit
    003406                                  .SPFUN  #XAREA,#1,R5,#0,#0,#0 ;Do the function
    003452  103404                          BCS     2$              ;Br if error on .SPFUN
    003454                                  .CLOSE  #1              ;Close CL channel
    003462  000207                          RETURN
    003464                          2$:     .PURGE  #1              ;Make sure channel is closed
    003472                                  FABORT  #EM$IUN         ;Say invalid unit number
                                    ;
                                    ;   SET CL PARITY  driver routine
                                    ;
    003502  004767  000000G         STCLPO: CALL    SKPSPC          ;Skip any spaces
    003506  122327  000075                  CMPB    (R3)+,#'=       ;Should have an equal sign
    003512  001401                          BEQ     1$
    003514  005303                          DEC     R3              ;Point to start of keyword
    003516  012704  001556'         1$:     MOV     #CLPRHD,R4      ;Point to driver table
    003522  004767  000000G                 CALL    SETWRD          ;Process the option
    003526  000207                          RETURN
```

```
                                    ;-------------------------------------------------------------------------------
                                    ;  SET CL PARITY={EVEN ODD NONE}
                                    ;
                                    ;  Inputs:
                                    ;   R2 = CL unit number index
                                    ;   R4 = Pointer to word with parity control flags
                                    ;
003530  011404                      STCLPR: MOV     (R4),R4         ;Get parity flags
003532  016202  000000G             STCLP1: MOV     CL$LIX(R2),R2   ;Get number of line CL unit is connected to
003536  001406                              BEQ     9$              ;Br if not assigned to a line
003540  116205  000001G                     MOVB    LMXPRM+1(R2),R5 ;Get current control flags for line
003544  042705  000000C                     BIC     #LP$PAR!LP$ODD,R5 ;Clear parity control flags
003550  050405                              BIS     R4,R5           ;Put in new flags
003552  000665                              BR      STCLS1          ;Go set value for line
003554                              9$:     FABORT  #EM$CLN         ;CL unit is not assigned to a line
                                    ;
                                    ;  SET CL NOPARITY
                                    ;
003564  005004                      STCLNP: CLR     R4              ;Clear parity flags
003566  000761                              BR      STCLP1          ;Go do the set
```

```
                                ;-----------------------------------------------------------------------
                                ;   SET CL BITS=n
                                ;
003570  016202  0000000         STCLBT: MOV     CL$LIX(R2),R2   ;Get index for line CL unit is assigned to
003574  001425                          BEQ     9$              ;Br if not assigned to a line
003576  004767  0000000                 CALL    ACRDEC          ;Accrue parameter value
003602  116205  0000010                 MOVB    LMXPRM+1(R2),R5 ;Get current flags for line
003606  042705  0000000                 BIC     #LP$7BT,R5      ;Assume 8 bits wanted
003612  020127  000007                  CMP     R1,#7.          ;7 bits wanted?
003616  001003                          BNE     1$              ;Br if not
003620  052705  0000000                 BIS     #LP$7BT,R5      ;Set 7 bit flag
003624  000407                          BR      2$
003626  020127  000010          1$:     CMP     R1,#8.          ;8 bits wanted?
003632  001404                          BEQ     2$              ;Br if yes
003634                                  FABORT  #EM$ICL         ;Invalid character length
003644  000167  177456          2$:     JMP     STCLS1          ;Go set value
003650                          9$:     FABORT  #EM$CLN         ;CL unit not assigned to line
```

```
                                  ;------------------------------------------------------------------------
                                  ;  SET CL LINE=n
                                  ;
  003660                          STCLLN:
                                  ;
                                  ;  Accrue the line number
                                  ;
  003660  004767  000000G                 CALL    ACRDEC            ;Accrue decimal value
                                  ;
                                  ;  Do the EMT to assign this CL unit to a line and check for errors.
                                  ;
  003664  006202                          ASR     R2                ;Convert CL unit index to unit #
  003666  010267  174152                  MOV     R2,CLAEMT+2       ;Set CL unit number
  003672  010167  174150                  MOV     R1,CLAEMT+4       ;Set time-sharing line number
  003676  012700  000042'                 MOV     #CLAEMT,R0        ;Point to EMT argument block
  003702  104375                          EMT     375               ;Do the EMT
  003704  103045                          BCC     20$               ;Br if no error
                                  ;
                                  ;  An error occurred on the EMT.
                                  ;  Print an error message
                                  ;
  003706  113702  000000G                 MOVB    @#ERRLOC,R2       ;Get EMT error code
  003712  120227  000002                  CMPB    R2,#2.            ;Invalid CL unit # ?
  003716  001004                          BNE     3$                ;Br if not
  003720                                  FABORT  #EM$IUN           ;Invalid CL unit number
  003730  120227  000003          3$:     CMPB    R2,#3.            ;Invalid line number?
  003734  001004                          BNE     4$                ;Br if not
  003736                                  FABORT  #EM$ILN           ;Invalid line number
  003746  120227  000004          4$:     CMPB    R2,#4.            ;Line already in use by a CL unit?
  003752  001004                          BNE     5$                ;Br if not
  003754                                  FABORT  #EM$ACL           ;Line already being used by CL
  003764  120227  000005          5$:     CMPB    R2,#5.            ;Is line in use by a time-sharing user?
  003770  001004                          BNE     6$                ;Br if not
  003772                                  FABORT  #EM$TSL           ;Line busy as time-sharing line
  004002  120227  000006          6$:     CMPB    R2,#6.            ;Is this CL unit currently busy?
  004006  001004                          BNE     20$               ;Br if not
  004010                                  FABORT  #EM$CLB           ;This CL unit is busy
                                  ;
                                  ;  Finished
                                  ;
  004020  000207                  20$:    RETURN
```

```
                                ;--------------------------------------------------------------------------
                                ;   SET CL VERSION=n
                                ;
 004022  004767  000000G        STCLVR: CALL    ACRDEC          ;Accrue parameter value
 004026  120127  000017                 CMPB    R1,#15.         ;Is this a valid argument
 004032  002403                         BLT     9$              ;Br if not
 004034  110167  000000G                MOVB    R1,CLVERS       ;Set version number
 004040  000404                         BR      10$             ;Return
 004042                         9$:     FABORT  #EM$IVN         ;Invalid version number
                                ;
                                ;   Finished
                                ;
 004052  000207                 10$:      RETURN
```

```
                                        ;-------------------------------------------------------------------------
                                        ;  SET CLn ENDSTRING='string'
                                        ;
 004054  005762  0000000                CLESTR: TST     CL$LIX(R2)      ;Is this CL unit assigned to a line?
 004060  001002                                 BNE     2$              ;Br if yes
 004062  000167  177102                         JMP     STCLNL          ;Br if not
 004066  010446                         2$:     MOV     R4,-(SP)
 004070  010546                                 MOV     R5,-(SP)
                                        ;
                                        ;  Accrue the string
                                        ;
 004072  004767  0000000                        CALL    ACRSTR          ;Accrue the string
                                        ;
                                        ;  Make sure the string is not too long
                                        ;
 004076  020027  0000000                        CMP     R0,#CLEOFS      ;Is string too long?
 004102  101046                                 BHI     10$             ;Br if yes
                                        ;
                                        ;  Move string over in BLKO to allow room for ENDPAGE value in 1st word
                                        ;
 004104  012705  0000000                        MOV     #BLKO,R5        ;Point to start of buffer
 004110  005200                                 INC     R0              ;Get length of string with null
 004112  060005                                 ADD     R0,R5           ;Point past last char in string
 004114  010504                                 MOV     R5,R4
 004116  062704  000002                         ADD     #2,R4           ;Add 1 word offset
 004122  114544                         3$:     MOVB    -(R5),-(R4)     ;Move string over
 004124  077002                                 SOB     R0,3$
 004126  012767  000377  0000000                MOV     #377,BLKO       ;Set value for ENDPAGE (377=no change)
                                        ;
                                        ;  Do lookup on CL unit
                                        ;
 004134  004767  000306                         CALL    CLLOOK          ;Do lookup on channel 1 to CL unit
                                        ;
                                        ;  Perform the .SPFUN to set the ENDSTRING
                                        ;
 004140                                         .SPFUN  #XAREA,#1,#CLSFEP,#BLKO,#10.,#0
 004204                                         .CLOSE  #1              ;Close the channel
                                        ;
                                        ;  Finished
                                        ;
 004212  012605                                 MOV     (SP)+,R5
 004214  012604                                 MOV     (SP)+,R4
 004216  000207                                 RETURN
                                        ;
                                        ;  String is too long
                                        ;
 004220                                 10$:    FABORT  #EM$STL         ;String too long
```

```
                                    ;---------------------------------------------------------------------
                                    ;  SET CLn TRANSLATE=(ext=int,ext=int,...)
                                    ;
004230  016201  000000G             CLTRNS: MOV     CL$LIX(R2),R1   ;Is this CL unit assigned to a line?
004234  001002                              BNE     2$              ;Br if yes
004236  000167  176726                      JMP     STCLNL          ;Br if not
004242  004767  000002              2$:     CALL    PRSTRN          ;Parse the translate qualifier
004246  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
                                    ;
004250  010446                      PRSTRN: MOV     R4,-(SP)
004252  010546                              MOV     R5,-(SP)
                                    ;
                                    ;  Get pointer to translation table area for this line
                                    ;
004254  016104  000000G                     MOV     LCXTBL(R1),R4   ;Get pointer to translation table for line
004260  001004                              BNE     7$              ;Br if there is one
004262                                      FABORT  #EM$TMT         ;No translation table allocated
                                    ;
                                    ;  See if equal sign and open paren follow keyword
                                    ;
004272  004767  000000G             7$:     CALL    SKPSPC          ;Skip over any spaces
004276  122327  000075                      CMPB    (R3)+,#'=       ;Does equal sign follow keyword?
004302  001406                              BEQ     1$              ;Br if yes
004304  126327  177777  000050              CMPB    -1(R3),#'(      ;Did user omit equal sign?
004312  001413                              BEQ     2$              ;Br if yes
004314  005303                              DEC     R3              ;Point back to character
004316  000447                              BR      3$              ;Clear the translation table
004320  004767  000000G             1$:     CALL    SKPSPC          ;Skip more spaces
004324  122327  000050                      CMPB    (R3)+,#'(       ;Should have open paren
004330  001404                              BEQ     2$              ;Br if got open paren
004332                                      FABORT  #EM$CSE         ;Invalid syntax
                                    ;
                                    ;  Begin to accrue each value pair
                                    ;
004342  005005                      2$:     CLR     R5              ;Count pairs in R5
                                    ;
                                    ;  See if we have reached the end of the list
                                    ;
004344  004767  000000G             5$:     CALL    SKPSPC          ;Skip any spaces
004350  122327  000051                      CMPB    (R3)+,#')       ;At end of the list?
004354  001430                              BEQ     3$              ;Br if yes
004356  005303                              DEC     R3              ;Point back to character
                                    ;
                                    ;  See if we have room for another value
                                    ;
004360  020527  000000G                     CMP     R5,#MXTTCT      ;Room for another value?
004364  103404                              BLO     4$              ;Br if yes
004366                                      FABORT  #EM$TMT         ;Too many translation values
```

```
004376  005205                  4$:     INC     R5              ;Count another value
                                ;
                                ;  Accrue the external value
                                ;
004400  004767  0000000                 CALL    ACROCT          ;Accrue the external value
004404  110164  000001                  MOVB    R1,1(R4)        ;Store in high-order byte of word
                                ;
                                ;  Accrue the internal value
                                ;
004410  004767  0000000                 CALL    ACROCT          ;Accrue the internal value
004414  110114                          MOVB    R1,(R4)         ;Store in low-order byte of word
004416  005724                          TST     (R4)+           ;Point to next word
004420  004767  0000000                 CALL    SKPSPC          ;Skip any spaces
004424  122327  000054                  CMPB    (R3)+,#',       ;Comma separator?
004430  001745                          BEQ     5$              ;Br if yes
004432  005303                          DEC     R3              ;Point back to char we skipped
004434  000743                          BR      5$              ;Loop and get rest of list
                                ;
                                ;  We have reached the end of the list
                                ;
004436  005014                  3$:     CLR     (R4)            ;Store a zero to terminate the list
                                ;
                                ;  Finished
                                ;
004440  012605                          MOV     (SP)+,R5
004442  012604                          MOV     (SP)+,R4
004444  000207                          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.
                                ;
004446  010246                  CLLOOK: MOV     R2,-(SP)
                                ;
                                ;  Construct CL device name
                                ;
004450  006202                          ASR     R2              ;Get cl unit number
004452  020227  000007                  CMP     R2,#7.          ;Should this be a CL or C1 unit?
004456  101405                          BLOS    2$              ;Br if CL
004460  162702  000010                  SUB     #8.,R2          ;Remove C1 unit bias
004464  066702  173326                  ADD     R50C10,R2       ;Add "C10" to form device name
004470  000402                          BR      1$
004472  066702  173316          2$:     ADD     R50CLO,R2       ;Add "CLO" to form device name
004476  010267  173302          1$:     MOV     R2,CLDEV        ;Save device name
004502                                  .LOOKUP #XAREA,#1,#CLDEV;Open channel to CL unit
004522  103402                          BCS     10$             ;Br if error on lookup
                                ;
                                ;  Finished
                                ;
004524  012602                          MOV     (SP)+,R2
004526  000207                          RETURN
                                ;
                                ;  Error on lookup to CL unit
                                ;
004530                          10$:    FABORT  #EM$IUN         ;Invalid unit number
```

```
                                              .SBTTL  .       Host
                                      ;-------------------------------------------------------------------------------
                                      ;   SET HOST/DTE=n
                                      ;   Cross connect time-sharint line with CL unit.
                                      ;
004540  004767  000000G               SETHST: CALL    CKTERM          ;Require TERMINAL privilege for SET HOST
                                      ;
                                      ;   Do command scanning
                                      ;
004544  012767  177777  173302                MOV     #-1,TTXCL+4     ;Initially set CL unit to -1
004552  012704  001604'                       MOV     #HOSTHD,R4      ;Point to option driver table
004556  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.
                                      ;
004562  016705  173266                        MOV     TTXCL+4,R5      ;Get CL unit number
004566  020527  000000G                       CMP     R5,#CLTOTL      ;Is this a valid unit number
004572  103404                                BLO     1$              ;Br if yes
004574                                        FABORT  #EM$IUN         ;Invalid CL unit number
004604  006305                        1$:     ASL     R5              ;Convert to CL index number
004606  010567  173242                        MOV     R5,TTXCL+4
004612  005765  000000G                       TST     CL$LIX(R5)      ;Is this CL unit connected to a line?
004616  001004                                BNE     2$              ;Br if yes
004620                                        FABORT  #EM$CLN         ;Not connected to line
004630  010500                        2$:     MOV     R5,R0           ;Get CL unit index
004632  004767  000000G                       CALL    CKCLUS          ;See if CL unit in use by another job
004636  006300                                ASL     R0              ;Any job using CL unit?
004640  001407                                BEQ     3$              ;Br if not
004642  120067  000000G                       CMPB    R0,CORUSR       ;Is it our job?
004646  001404                                BEQ     3$              ;Br if yes
004650                                        FABORT  #EM$CLB         ;This CL unit is busy
                                      ;
                                      ;   Perform the connection
                                      ;
004660  116701  000000G               3$:     MOVB    CORUSR,R1       ;Get our job index number
004664  016761  173110  000000G               MOV     R50HST,LPRG1(R1);Set name of running program to "$HOST$"
004672  016761  173104  000000G               MOV     R50HST+2,LPRG2(R1)
004700  012700  000050'                       MOV     #TTXCL,R0       ;Point to EMT argument block
004704  104375                                EMT     375             ;Make the cross connection
                                      ;
                                      ;   Print message saying cross connection broken
                                      ;
004706  016761  000000G 000000G               MOV     R50KMN,LPRG1(R1);Reset program name to KMON
004714  016761  000002G 000000G               MOV     R50KMN+2,LPRG2(R1)
004722                                        .PRINT  #TM$XBK
004730  000167  000000G                       JMP     RDCMD           ;Finished with command
                                      ;
                                      ;   Process the /PORT=ddn qualifier
                                      ;
004734  010246                        HSTPRT: MOV     R2,-(SP)
004736  004767  000000G                       CALL    CHKEQ           ;Equal sign should follow qualifier name
004742  004767  000000G                       CALL    GTRD50          ;Accrue the device name
004746  122327  000072                        CMPB    (R3)+,#':       ;Colon specified with device name?
004752  001401                                BEQ     1$              ;Br if yes
004754  005303                                DEC     R3              ;Backup pointer
004756  016700  000000G               1$:     MOV     R50BUF,R0       ;Get accrued device name
```

```
   004762  004767  000000G                  CALL    ASNSRC              ;See if this is a logical device name
   004766  103402                           BCS     2$                  ;Br if not logical name
   004770  016200  000000G                  MOV     AT$DEV(R2),R0       ;Get physical device name
   004774  004767  000000G          2$:     CALL    CHKCLU              ;Convert CL and C1 names to unit numbers
   005000  103404                           BCS     10$                 ;Br if not CL or C1 unit
   005002  010067  173046                   MOV     R0,TTXCL+4          ;Set CL unit number
   005006  012602                           MOV     (SP)+,R2
   005010  000207                           RETURN
   005012                           10$:    FABORT  #EM$CLX             ;Port unit must be CL or C1 unit
                                    ;
                                    ;  Process the /DTE=n SET HOST qualifier
                                    ;
   005022  010146                   HSTDTE: MOV     R1,-(SP)
                                    ;
                                    ;  Accrue the CL unit number
                                    ;
   005024  004767  000000G                  CALL    ACRDEC              ;Accrue the CL unit number
   005030  010167  173020                   MOV     R1,TTXCL+4          ;Store CL unit index in EMT arg block
   005034  012601                           MOV     (SP)+,R1
   005036  000207                           RETURN

           000001                           .END
Errors detected:  0

*** Assembler statistics


Work  file  reads: 0
Work  file writes: 0
Size of work file: 11896 Words  ( 47 Pages)
Size of core pool: 17920 Words  ( 70 Pages)
Operating  system: RT-11

Elapsed time: 00:01:28.06
DK:TSKST2,LP:TSKST2=DK:TSKST2.MAC/C/N:SYM
```

```
$JSTLG   1-73
$8BIT    1-111     5-19      5-20      5-46      5-72      5-73      5-74
$AUTO    1-87      5-7       16-6      19-5
$CARUP   1-85
$CCLRN   1-86
$CFABT   1-106
$CFALL   1-112
$CFCCL   1-112
$CFDCC   1-112
$CFOPN   1-118
$CFSOT   1-110
$CHACT   1-61
$CLTST   1-96
$CTRLC   1-104
$CTRLD   1-154
$CTRLO   1-61
$CTRLS   1-91
$DBKMN   1-84
$DEAD    1-158     5-9       5-10      15-14
$DEBUG   1-155
$DEFER   1-124     5-12      5-13
$DETCH   1-89
$DIBOL   1-73
$DILUP   1-108     7-16      14-47     15-8      19-6      23-137
$DISCN   1-90
$DOOFF   1-114
$DUPRN   1-109
$ECHO    1-111     5-17      5-18
$EMTTR   1-95
$FORM    1-110     5-21      5-22
$FORMO   1-112     5-23      5-24
$HARD    1-158
$HITTY   1-72
$INCOR   1-128
$INDAB   1-159
$INDDF   1-157
$INDRN   1-157
$INIT    1-158
$INKMN   1-104
$KED     1-128
$KINIT   1-68
$LC      1-111     5-32      5-33
$LOFCF   1-203
$MLOCK   1-77
$NOIN    1-72
$NOINT   1-204
$NOWTT   1-72      23-125    23-131
$PAGE    1-111     5-35      5-36
$PHONE   1-158     5-39      5-40
$PRGLK   1-87
$QTSET   1-133     23-89     23-106
$QUIET   1-125     23-90     23-107
$RNIOP   1-205
$SCOPE   1-111     5-44      23-118
$SGALL   1-124
$SGHIO   1-123
```

Cross reference table (CREF V05.04)

```
$SQIIO   1-123
$SQQ0    1-105
$SQQ1    1-123
$SQQ1A   1-123
$SQQ1B   1-123
$SQQ1C   1-123
$SQQ2    1-123
$SQQ3    1-105
$SLKED   1-82
$SLLET   1-202
$SLON    1-202     23-16     23-40     23-62     23-70     23-79     23-119
$SLTTY   1-202
$SNWTT   1-155     23-124    23-130
$SPLJB   1-91
$STSNG   1-61       5-47      5-48
$SUCF    1-113
$SYSPS   1-160      5-51      5-52      5-53      5-54      5-55      5-56
$TAB     1-110      5-57      5-58
$TAPE    1-153      5-59      5-60
$TECO    1-133
$TTGAG   1-118      5-25      5-26
$UCLCF   1-119
$UCLCL   1-121
$UCLCM   1-121
$UCLRN   1-120
$UKMON   1-85
$VNOTT   1-48
$VTESC   1-81
$WILD    1-134
...V1    8-25       8-28     23-93     23-96     23-110    23-113    24-128    24-130    24-130    24-132    29-36     29-37
         29-37     31-22     31-22     31-22     32-46
...V2    24-128    24-128    24-128    24-128#   24-128#   24-128#   24-130    24-130#   24-132    24-132#   29-36     29-36
         29-36     29-36#    29-36#    29-36#    29-37     29-37#    31-22     31-22     31-22#    31-22#
ABRTAD   1-48
ABRTCD   1-48
ABRTCF   1-165
ACRDEC   1-173      7-8       8-9      14-28     21-7      24-73     26-6      27-8      28-4      32-74
ACRFIL   1-172
ACRFN    1-165
ACROCT   1-175     30-62     30-67
ACRSPD   1-76      16-5      24-97
ACRSTR   1-37       9-23     29-12
ACRTXT   1-39
AD$$SZ   1-64
AD$DVU   1-64
AD$JOB   1-64
ADM3A    1-148     23-69
ADM3FL   1-149     23-68
ADM3NO   1-151     23-67
AF$DBG   1-204
AF$HIE   1-204
AF$IOP   1-205
AF$MEM   1-47
AF$NOI   1-204
AF$NOW   1-47
AF$PLK   1-204
```

```
AF$SCA   1-47
ALCDEV   1-30
ALCEND   1-63
ALCHD1   1-60
ALCHD2   1-60
ALCTBL   1-63
ALDBLK   1-188
ALDEX    1-187     1-188
ALFN     1-199
AMBOPT   1-173     1-184
AR$$SZ   1-202
AR$CNT   1-201
AR$CON   1-201
AR$CPH   1-201
AR$CPL   1-201
AR$DMY   1-202
AR$PRG   1-201
AR$PRJ   1-201
AR$UNM   1-201
ARNRPB   1-202
ASDEX    1-170
ASKLNM   1-169
ASNHD1   1-182
ASNHD2   1-182
ASNOVF   1-170     1-195
ASNSRC   1-137    32-58
AT$DEV   1-56     32-60
AUTHFN   1-192
BADCMD   1-169
BADPMT   1-176
BADPRI   1-176     8-22
BADSAV   1-169
BASMAP   1-142
BDFNAM   1-172
BDLGOP   1-177
BDLIN    1-190
BELL     2-7#
BLANK    2-6#
BLKO     1-168     9-28     29-21     29-28*    29-36
BLKWDS   2-11#
BOTDEV   1-103
BOTUNI   1-103
BS01    16-16     17-10     18-14     19-10     23-27     23-29#    23-97     23-114    23-120    23-147    23-155
BS02    23-9      23-20#    23-41     23-48     23-55     23-63     23-71     23-80
C. CSW   1-115
C. DEVQ  1-115
C. SBLK  1-115
C. USED  1-41
CALUCL   1-181
CASCBR   1-97
CASCUP   1-97
CASTBR   1-97
CASTBW   1-97
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$TOP  1-133
CDBUF   1-38
CDGET   1-39
CF$IND  1-159
CF$QUT  1-159    23-95    23-112
CFABLV  1-160
CFBLK   1-125
CFBUF   1-94
CFCHAN  1-124
CFEND   1-94
CFHOLD  1-132
CFIND   1-114
CFLFL4  1-119
CFNEST  1-117
CFPNT   1-125
CFSEND  1-118
CFSP    1-118
CFSPND  1-131
CFSTK   1-68
CFSTS   1-159    23-93    23-96    23-110   23-113
CHAIN   1-104
CHKALC  1-111
CHKCLU  1-36     32-61
CHKDEV  1-196
CHKDLM  1-186
CHKEQ   1-45      9-9     32-52
CHKMNT  1-171
CHKMTX  1-171
CHKTTD  1-194
CHNSIZ  1-41
CINDAT  1-141
CINFLG  1-48
CKACOJ  1-45     12-11    13-19
CKCLUS  1-37     32-27
CKPRIV  1-173    15-4
CKSYPV  1-45
CKTERM  1-174    14-40    16-4     17-4     18-7     19-4     20-5     21-4     22-8     22-16    23-136   23-143
       23-151    24-8     32-6
CL$COL  1-66
CL$EPN  1-41      5-74
CL$EPS  1-41
CL$LEN  1-101     5-110
CL$LIN  1-101    24-90*
CL$LIX  1-62     24-57    24-64    24-71    24-81    24-88    24-95    24-125   25-9     26-4     29-4     30-4
       32-23
CL$OPT  1-100    24-59*   24-66*
CL$RQH  1-63
CL$SKP  1-101     5-131
CL$WID  1-101     5-132   24-83*
```

```
CL$WQH   1-63
CL$XLN   1-37
CLAEMT   4-24#     27-13*     27-14*     27-15
CLCLOP   5-91       5-93       5-97       5-99       5-101      5-105      5-107      5-109      5-112      5-114      5-117      5-119
         5-122      5-123      5-124      5-127     24-64#
CLDEV    4-5#      31-21*     31-22
CLDEVX   1-139
CLEOFS   1-41      29-16
CLESTR   5-95      29-4#
CLFREE   1-57
CLLINE   1-57
CLLOOK  24-127     29-32      31-10#
CLOPHD   5-88#     24-31
CLOTIR   1-65
CLPRHD   5-125      5-140#    24-141
CLRDTR   5-16      18-6#
CLRPRV   1-43       6-14
CLRTTB   5-13       5-18       5-20       5-22       5-24       5-26       5-33       5-36       5-46       5-48       5-58       5-60
         5-73       5-74      23-152#
CLRTTP   5-7        5-10       5-40       5-52       5-54       5-56      23-151#
CLSFCH   1-69      24-115
CLSFEP   1-42      29-36      29-36
CLSFRS   1-34      24-120
CLSFSP   1-98
CLSPFN  24-116     24-125#
CLTOP    5-128     24-88#
CLTOTL   1-98      24-9       32-18
CLTRNS   5-129     30-4#
CLUNIT   1-57
CLVERS   1-57      28-7*
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$8BT   1-115      5-120      5-121      5-122      5-123      5-124      5-127
CO$BNI   1-100      5-116      5-117
CO$BNO   1-100      5-118      5-119
CO$CR    1-99       5-90       5-91
CO$CTL   1-99       5-108      5-109
CO$DEF   1-66
CO$DTR   1-98       5-92       5-93
CO$FF    1-99       5-96       5-97       5-98       5-99
CO$FFO   1-99       5-100      5-101
CO$LC    1-99       5-104      5-105
CO$LFI   1-100      5-111      5-112
CO$LFO   1-100      5-113      5-114
CO$TAB   1-99       5-106      5-107
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      23-85      23-91      23-102     23-108     32-30      32-36
CPFAUT    4-14#      5-163     11-29
CPFLAG    4-8#       6-13*      9-65*     11-29      13-8       13-25
CPFRES    4-13#      5-162     13-8       13-25
CPFSUS    4-12#      5-161     13-8
CPNAME    4-19#      6-11*      9-10       9-41       9-48      10-9       10-21
CPOPHD    5-156#     6-18
CPPFLG    5-161      5-162      5-163      7-65#
CPPID     4-17#      6-10*      7-18*     10-14      11-22      12-10      13-13
CPPNAM    5-159      9-4#
CPPPID    5-158      7-4#
CPPPRI    5-160      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$RDN    1-80
CSHALC    1-103
CSHDEV    1-108
CSHDVN    1-108
CSHHD     1-163
CSHMSG    1-203
CSHSIZ    1-143
CSIMS1    1-175     24-41
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$50H    1-113
CW$PRO    1-62
CXTBAS    1-107
CXTPAG    1-79
CXTRMN    1-41
CXTWDS    1-107
DATTIM    1-29
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    23-77
DIABLO   1-147    23-78
DIABNO   1-126    23-76
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   4-47#    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$ACL   1-54     27-31
EM$CAP   1-44     11-63
EM$CIP   1-53
EM$CLB   1-54     27-37     32-32
EM$CLN   1-53     24-53     25-15     26-17     32-25
EM$CLU   1-34     15-22
EM$CLX   1-36     32-66
EM$CNO   1-44     10-34
EM$CPO   1-44     11-59
EM$CSE   1-70     24-45     30-40
EM$HNI   1-68
EM$ICL   1-52     21-15     26-15
EM$IDR   1-42     13-38
EM$ILN   1-53      1-54      7-27     14-36     16-20     17-11     18-15     27-28
EM$IST   1-42
EM$IUN   1-53     24-11     24-111    24-133    27-25     31-32     32-20
EM$IVN   1-53     28-9
EM$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$PTA     1-36
EM$PTU     1-36
EM$SLT     1-54
EM$SLW     1-54
EM$SPL     1-39
EM$STL     1-42      9-55     29-47
EM$TMT     1-34     30-26     30-57
EM$TSL     1-54     27-34
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     27-22
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    23-139     24-11     24-41     24-45     24-53    24-111    24-133     25-15
          26-15     26-17     27-25     27-28     27-31     27-34     27-37      28-9     29-47     30-26     30-40     30-57
          31-32     32-20     32-25     32-32     32-66
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     32-53
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     23-61
HAZLFL     1-77     23-60
HAZLNO     1-77     23-59
HIMAP     1-142
HIPRI     1-176
HNBUF     1-174
HOSTHD   5-148#     32-11
HSTDTE    5-149     5-150    32-70#
HSTPRT    5-151    32-51#
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   23-20*
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
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#     5-80      5-80#     5-81      5-81#
         5-82      5-82#     5-89      5-89#     5-90      5-90#     5-91      5-91#     5-92      5-92#     5-93      5-93#
         5-94      5-94#     5-95      5-95#     5-96      5-96#     5-97      5-97#     5-98      5-98#     5-99      5-99#
         5-100     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#
         5-141     5-141#    5-142     5-142#    5-143     5-143#    5-149     5-149#    5-150     5-150#    5-151     5-151#
         5-157     5-157#    5-158     5-158#    5-159     5-159#    5-160     5-160#    5-161     5-161#    5-162     5-162#
         5-163     5-163#
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     30-24
LD$RON   1-121
LDBASE   1-122
LDCLEN   1-71
LDDEVX   1-124
LDFLAG   1-122
LDMNT    1-70
LDNAM    1-169
LDNAME   1-122
LDOPHD   1-177
LDPDEV   1-122
LDSIZE   1-122
LF       2-4#
LF$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     24-98     25-11     26-7
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$7BT   1-52     21-8      21-11     26-8      26-11
LP$ODD   1-52      5-81      5-142    22-10     22-18     25-12
LP$PAR   1-52      5-80      5-81      5-141     5-142    22-10     22-18     25-12
LP$SPD   1-52     16-8      24-99
LPRG1    1-145    32-37*    32-44*
LPRG2    1-145    32-38*    32-45*
LPRI     1-160    12-17*
LPROG    1-90
LPROJ    1-90
LRBFIL   1-114
LRDTIM   1-80
LSCCA    1-110
LSECPT   1-95
LSPEMT   4-36#    16-10*    16-11*    16-12*    16-13     24-101*   24-103*   24-104
LSTACT   1-78
LSTATE   1-140
LSTDL    1-89
LSTHL    1-62
LSTIOL   1-62
LSTMX    1-156
LSTPL    1-135    14-34
```

```
TSKST2 -- Keyboard SET Command  MACRO V05.04  Monday 21-Dec-87 12:58 Page S-12
Cross reference table (CREF V05.04)


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      23-137
LSW2     1-104     5-7       5-12      5-13      5-17      5-18      5-19      5-20      5-21      5-22      5-32      5-33
         5-35      5-36      5-39      5-40      5-44      5-46      5-51      5-52      5-53      5-54      5-55      5-56
         5-57      5-58      5-59      5-60      5-72      5-73      5-74      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*
        23-67*    23-68*    23-76*    23-77*    23-89*    23-106*   23-118*
LSW2S    1-109    23-29*
LSW3     1-109     5-9       5-10      15-14
LSW4     1-127     5-23      5-24     23-90*    23-107*
LSW5     1-87     23-125*   23-131*
LSW6     1-155     5-47      5-48     23-124*   23-130*
LSW7     1-159     5-25      5-26     23-16*    23-40*    23-62*    23-70*    23-79*    23-119*
LSW8     1-123
LSW9     1-85
LTRMTP   1-148    23-8*     23-15*    23-28     23-39*    23-47*    23-54*    23-61*    23-69*    23-78*
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    23-93     23-96     23-110    23-113
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      30-55
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      5-79#     5-80      5-80
          5-81      5-81      5-82      5-82      5-88#     5-89      5-89      5-90      5-90      5-91      5-91      5-92
          5-92      5-93      5-93      5-94      5-94      5-95      5-95      5-96      5-96      5-97      5-97      5-98
          5-98      5-99      5-99      5-100     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     5-140#    5-141     5-141     5-142     5-142     5-143     5-143     5-148#    5-149
          5-149     5-150     5-150     5-151     5-151     5-156#    5-157     5-157     5-158     5-158     5-159     5-159
          5-160     5-160     5-161     5-161     5-162     5-162     5-163     5-163
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
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$RON   1-153
OTHRON   1-194
OTRMNT   1-196
OVRCOR   1-168
PO$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$ULN   1-35
PARHD    5-79#    20-10
PASLIN   1-134
PAUMSG   1-164
PBFEND   1-118
PEKADR   1-38
PEKEMT   1-38
PEKSIZ   1-38
PF$IOW   1-155
PF$SYS   1-155
PFCO     1-43      4-61    11-11
PFSO     1-43      4-54    11-10
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
PRIVC2   1-45
PRIVSO   1-43
PRMBUF   1-131
```

```
PRMEND   1-131
PRMPNT   1-130
PROSLT   1-52
PRSTRN  22-26      30-7       30-19#
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       5-157
PUSHCF   1-166
PVCEMT   4-59#     11-33#     11-45
PVNPW    1-43      11-12
PVON     1-87
PVSEMT   4-52#     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     32-57
R50C10   4-7#      31-18
R50CLO   4-6#      31-20
R50COM   1-110
R50DIR   1-166
R50DK    1-199
R50DSK   1-172
R50DUP   1-167
R50HST   4-4#      32-37      32-38
R50IND   1-166
R50K52   1-167
R50KED   1-167
R50KEX   1-167
R50KMN   1-37      32-44      32-45
R50LD    1-172
R50LDO   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$$SZ   1-51
RC$AEP   1-50
RC$AGE   1-50
RC$CNT   1-49
RC$EXC   1-50
RC$FLG   1-50
RC$GBL   1-50
RC$LCG   1-52
RC$LEN   1-51
RC$NAM   1-50
RC$OWN   1-49
RC$PVT   1-51
RC$USE   1-50
RCBBAS   1-51
RCBEND   1-51
RDB      1-139
RDBEND   1-139
RDCMD    1-164     5-28      5-29      5-34      6-30      14-60     23-87     23-104    24-49     32-47
RDERM    1-168
REMTXT   1-57
RESDEV   1-153
RJEMT    1-37     13-24
RNMS     1-179
RONTXT   1-184
RSR      1-156
RSTPRV   1-44
RSTQUT   5-42     23-103    23-106#
RT$$SZ   1-139
RT$NAM   1-139
RUNDEV   1-107
RUNEMT   1-168
RUNHD    1-164
RUNMS    1-200    23-139
S$HICP   1-92
S$INWT   1-93
S$IOFN   1-88
S$IOWT   1-145
S$LOW    1-90
S$MSWT   1-94
S$OTFN   1-88
S$OTLO   1-88
S$OTWT   1-93
S$QMIO   1-86
S$QUSR   1-145
S$RT     1-90
S$SFWT   1-93      1-145
S$SPCB   1-146
S$SPDB   1-146
S$SPND   1-87
```

```
S$TMWT   1-93
S$TTFN   1-88
S$TTSC   1-89
S$TWFN   1-88
S9600    1-93     19-8
SADM3A   5-5      23-67#
SAUTO    5-6      19-4#
SBPSUF   1-36
SCHAIN   1-124
SCNOPS   1-37     14-56     32-12
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    5-14     23-76#
SDNAME   1-140
SDSFCB   1-119
SDSKIP   1-127
SEARCH   1-164    24-32
SERFLG   1-61
SETCL    1-26     24-8#
SETDED   5-9      23-136#
SETDTR   5-15     18-4#
SETHD    1-173
SETHST   1-26     32-6#
SETJMP  23-31     23-33#
SETNSC   5-45     23-118#
SETPAR   5-37     20-4#
SETPRC   1-27      6-6#
SETPRM   4-16#    14-9*     14-41*    23-26     23-30
SETQUT   5-41     23-86     23-89#
SETTNW   5-70     23-130#
SETTTB   5-12      5-17      5-19      5-21      5-23      5-25      5-32      5-35      5-44      5-47      5-57      5-59
         5-72     23-138    23-144#
SETTTP   5-39      5-51      5-53      5-55     23-143#
SETTTY   1-27     14-5#
SETTW    5-69     23-124#
SETWRD   1-42     20-11     24-142
SETXON   5-71     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     24-137    30-30     30-37     30-48     30-70
SLA120   5-30     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
SPLHD    1-186
SPLHLA   1-177
SPLPND   1-194
SPSNG    1-187     1-189
SPUBUF   1-69
SPWFM    1-187     1-188
SQUME    5-43     23-75#
SRTSIZ   1-143
SRTSMS   1-194
SRTTXT   1-197
SSRMAP   1-197
STAEMT   4-66#    15-26#    15-27
```

```
STCLBT   5-89      26-4#
STCLEN   5-8       21-4#
STCLGR   5-102      5-103    24-81#
STCLLN   5-115     27-4#
STCLNL  24-53#     24-58    24-65     24-72     24-82     24-89     24-96     24-126    29-6      30-6
STCLNP   5-126     25-17#
STCLOP   5-90       5-92     5-96      5-98      5-100     5-104     5-106     5-108     5-111     5-113     5-116     5-118
         5-120      5-121    24-57#
STCLP1  25-9#      25-20
STCLPO   5-125     24-137#
STCLPR   5-141      5-142    5-143    25-8#
STCLRS   5-130     24-120#
STCLS1  24-101#    25-14    26-16
STCLSP   5-133     24-75#
STCLVL   5-94       5-110    5-131     5-132    24-71#
STCLVR   5-134     28-4#
STCLXN   5-135     24-115#
STERM   23-48#
STHAZL   5-27      23-59#
STLA36   5-11       5-31     23-13#
STLGCN   1-29
STLGHD   1-176
STNOPR   5-38      22-16#
STNOVR   1-27      23-85#
STPAR    5-80       5-81     5-82     22-8#
STPASK   1-194
STPFLG   1-92
STSP1   16-10#     19-9     21-17     22-12     22-19
STTRNS   5-61      22-24#
STTTSP   5-49      16-4#
STTTST   5-50      15-4#
STVRFY   1-27      23-102#
SUBARO   1-184
SUBTXT   1-197
SUCS     1-144
SUM1     1-193
SUM2     1-193
SUM3     1-193
SUM4     1-193
SUM5     1-194
SUM6     1-194
SUM7     1-194
SUMS     1-144
SUPCOD   1-144
SVT100   1-27       5-62     23-45#
SVT200   1-27       5-63     5-64      5-65      5-66     23-52#
SVT50    1-27       5-67     5-68     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     32-46
TMIDLH   1-71
TMIOH    1-71
TMIOWH   1-70
TMSWPH   1-71
TMSWTH   1-71
TMTOTH   1-70      1-193
TMTOTL   1-70      1-193
TMUSRH   1-70
TOTMMS   1-197
TOTON    1-92
TOTXT    1-176     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     4-30#     32-10*    32-17     32-22*    32-39     32-63*    32-75*
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
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
VT2008    1-146
VT20FL    1-150     23-53
VT20NO    1-150     23-52
VT52      1-147     23-8
VT52FL    1-148     23-7
VT52NO    1-126     23-6
```

```
WILDFL   1-72
WINSTT   4-72#    23-24
WLDNAM   2-12#
XAREA    1-165    23-93     23-96     23-110    23-113    24-128    29-36     31-22
XONEMT   4-42#    17-5*     17-6*     17-7
YESTXT   1-178
ZCLR     1-156
```

```
Cross reference table (CREF V05.04)

...CM1   24-128     29-36      31-22
...CM2   23-93      23-96      23-96      23-110     23-113     23-113     24-128     24-128     24-128     24-128     24-128     24-128
         29-36      29-36      29-36      29-36      29-36      31-22      31-22
...CM3   24-130     24-132     29-37
...CM5    8-25       8-28      23-93      23-96      23-110     23-113     24-128     29-36      31-22      32-46
...CM6   23-93      23-96      23-110     23-113
.CLOSE    1-20#     24-130     29-37
.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#     23-93      23-110
.HERR     1-22#
.LOOKU    1-20#     31-22
.PRINT    1-20#      8-25       8-28      32-46
.PURGE    1-17#     24-132
.PVAL     1-22#     23-96      23-113
.READW    1-17#
.REOPE    1-18#
.SAVES    1-18#
.SERR     1-22#
.SPFUN    1-19#     24-128     29-36
.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       5-80
          5-81       5-82       5-89       5-90       5-91       5-92       5-93       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      5-141      5-142      5-143      5-149      5-150      5-151      5-157      5-158      5-159      5-160      5-161
          5-162      5-163
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      23-139     24-11      24-41      24-45      24-53      24-111     24-133     25-15
         26-15      26-17      27-25      27-28      27-31      27-34      27-37      28-9       29-47      30-26      30-40      30-57
         31-32      32-20      32-25      32-32      32-66
FERR      3-4#       8-22       9-55
FWARN     3-23#
TBLDEF    3-35#      5-4        5-79       5-88       5-140      5-148      5-156
TBLEND    3-61#      5-75       5-83       5-136      5-144      5-152      5-164
```