

Table of contents

6-	1	SET commands
6-	2	Process
14-	1	Terminal
24-	1	CL
32-	1	Host

```

1          .TITLE  TSKST2 -- Keyboard SET Command routines
2          .ENABL  LC
3          .DSABL  QBL
4 000000   .CSECT  TSKST2
5 000000   TSKST2:
6          ;
7          ; TSKST2 is the portion of TSKMON that contains the code
8          ; to implement the SET command.
9          ;
10         ; Copyright 1978, 1979, 1980, 1981, 1982, 1983, 1984, 1985.
11         ; S&H Computer Systems, Inc.
12         ; Nashville, Tennessee
13         ;
14         ; Macro calls
15         ;
16         .MCALL  .CSISPC, .TTOUTR, .SRESET
17         .MCALL  .READW, .TTYIN, .TTYOUT, .PURGE
18         .MCALL  .CSIQEN, .SAVEST, .REOPEN
19         .MCALL  .OTLIN, .OTIN, .DATE, .SPFUN
20         .MCALL  .PRINT, .CLOSE, .LOOKUP
21         .MCALL  .WRITW, .ENTER, .EXIT
22         .MCALL  .SERR, .HERR, .FPROT, .QVAL, .PVAL
23         ;
24         ; Global definitions
25         ;
26         .GLOBL  SETCL, SEHGT
27         .GLOBL  SETPRC, SETTTY, STVRFY, STNOVR, SVT50, SVT100, SVT200
28         .GLOBL  TSKST2, CMDHD, CMDOFF, KDOCIN, SKPSPC, UCLCMD
29         .GLOBL  DORUN, CMDFRM, CMDDSN, STLGCN, DATTIM, PRGALL
30         .GLOBL  DLCEMT, ALCDEV, CMDSHO, CMDSET, CMDWHO, CMDMEM, CMDUSE
31         ;
32         ; Global references
33         ;
34         .GLOBL  EM$LAS, EM$LFD, EM$CLU, LWINDO, CLSFRS, LCXTBL, MXTTCT, EM$TMT
35         .GLOBL  PA$GRC, PA$UKC, PA$DSC, PA$BLD, PA$ULN, PA$DWD, PA$HQL, PA$LET
36         .GLOBL  EM$PTA, EM$PTU, SBPSUF, CHKCLU, EM$CLX, PA$BEL, EM$OPR, PA$NWD
37         .GLOBL  SCNOPS, TM$XBK, CL$XLN, CKCLUS, ACRSTR, R50KMN, SJEMT, RJEMT
38         .GLOBL  PEKEMT, PEKADR, PEKSIZ, TM$NNR, CDBUF, CDGET, TM$IN1, TM$IN2
39         .GLOBL  SYPSWD, EM$SPL, ACRTXT, JPWDEV, JPWTYP, JPWFLG
40         .GLOBL  EM$WCO, EM$WC1, EM$WC2, EM$WC3, EM$WCM, P2$CGR
41         .GLOBL  CXTRMN, R$CHN, R$XCHN, CHNSIZ, C. USED, CL$EPN, CL$EPS, CLEOFS
42         .GLOBL  EM$STL, EM$IST, CLSFEP, EM$IDR, VPRIDF, SETWRD
43         .GLOBL  CLRPRV, OPTLST, PFSO, PFCO, PVNPW, P0$SPV, PRIVSO, PRVOPT
44         .GLOBL  TM$PVA, TM$PVC, PRIVAO, EM$CNO, EM$CPO, EM$CAP, RSTPRV
45         .GLOBL  CHKEQ, CKACDJ, P0$OPR, CKSYPV, P2$TRM, PRIVC2, P0$NAM, EM$NPR
46         .GLOBL  INSTBL, INGADR, INGEMT, IIBUF, II$NAM, II$FLG, II$SZ, EM$NAD
47         .GLOBL  INSTBN, AF$SCA, AF$NOW, AF$MEM, P0$DBG, PRIVCO, PRVLST
48         .GLOBL  ABRTAD, ABRTCD, CINFLG, $VNOTT, II$PRV, II$NPV
49         .GLOBL  TM$RD1, TM$RD2, TM$LCL, TM$GBL, SPACE1, RC$OWN, RC$CNT
50         .GLOBL  RC$EXC, RC$AGE, RC$AEP, RC$USE, RC$FLG, RC$GBL, RC$NAM
51         .GLOBL  RC$LEN, RC$PVT, RCBBAS, RCBEND, RC$SZ, SHRRCB, SHRRCN
52         .GLOBL  LP$SPD, LP$PAR, LP$ODD, LP$7BT, EM$ICL, PROSLT, RC$LCO
53         .GLOBL  EM$NPD, EM$IILN, EM$CIP, EM$NSF, EM$IUN, EM$CLN, EM$IVN
54         .GLOBL  EM$IILN, EM$ACL, EM$TSL, EM$CLB, EM$NSL, EM$SLT, EM$SLW
55         .GLOBL  SLKDON, SLKDOF, EM$UIO, TM$PR1, TM$PR2, TM$LPR, TM$HPR
56         .GLOBL  TM$HPE, TM$CNO, TM$CDS, TM$CEN, TRMHD1, TRMHD2, AT$DEV
57         .GLOBL  OPRTXT, CLLINE, LCLTXT, REMTXT, TM$AUT, CLFREE, CLUNIT, CLVERS

```

58 . GLOBL TM#CL0, TM#CL1, TM#CL2, TM#CL3, TM#CL4, TM#CL5, TM#CL6
59 . GLOBL QHDMS1, QHDMS2, DVSHH1, DVSHH2, DVSHH3, SYASHD, DKASHD
60 . GLOBL TM#NAD, ALCHD1, ALCHD2, TM#NSD, TM#SDN, LNAME
61 . GLOBL CORUSR, LSW, #CTRLD, SERFLG, IDABFL, #CHACT, #STSNQ
62 . GLOBL LSTHL, LCLUNT, FSTIOL, LSTIOL, CL#LIX, CW#PRO, CONFG2
63 . GLOBL CL#RQH, CL#WQH, MAXALC, ALCTBL, ALCEND
64 . GLOBL AD#DVU, AD#JOB, AD##SZ, UCIDEF, HANCHN
65 . GLOBL NEDCHR, LOU'IR, LINIR, LINRTS, CLOTIR
66 . GLOBL CO#DEF, CL#COL, LCDTYP, SOPALC, SOPDAT, SOPTIM
67 . GLOBL UTRPAD, JSWLOC, ERRLOC, MAXMEM, MAXPRI
68 . GLOBL USRSTK, #KINIT, CFSTK, MXJMEM, DFJMEM, EM#HNI
69 . GLOBL SPUBUF, SXBPNT, MXJADR, CLSFCH
70 . GLOBL TMTOTH, TMTOTL, TMUSRH, TMIOWH, LDMNT, EM#CSE
71 . GLOBL TMSWTH, TMIDLH, TMIOH, TMSWPH, LDCLEN
72 . GLOBL WILDFL, #NOIN, #NOWTT, #HITTY
73 . GLOBL TECO, EDIT, KED, K52, \$1STLG, \$DIBOL
74 . GLOBL SH#VAL, SH#NAM, SH##SZ, SH#RTN, SH#FLG
75 . GLOBL SO#NVL, SO#OCT, SO#NO, HANENT, HANSIZ
76 . GLOBL H. CSR, H. VEC, DVSTAT, SFID, ACRSPD, HANPAR, LSTSPL
77 . GLOBL HAZEL, HAZFL, HAZLN, #MLOCK, MDT, GETKCH
78 . GLOBL LINBUF, LINNXT, LSTACT, PRGTOP, PRGSIZ, KMNHI
79 . GLOBL KMNTOP, KMNPOS, KMNSTK, KMNSTR, CXTAG, FSTIOL
80 . GLOBL LINPNT, LINCNT, LACTIV, LRDTIM, CS#RON
81 . GLOBL LOTBUF, LOTNXT, LOTPNT, #VTESE
82 . GLOBL LOTSIZ, LOTSPC, LCOL, #SLKED, ESC, VDBFLG
83 . GLOBL LAFSIZ, LFWLIM, LINCUR, NUMON, ILSW2
84 . GLOBL #DBKMN
85 . GLOBL #CARUP, DOAGN, UKMNAM, #UKMON, LSW9
86 . GLOBL LSUCF, #CCLRN, VLDSYS, EM#NUK, S#QMIO
87 . GLOBL KL3CLR, #PRGLK, LSW5, PVON, S#SPND, #AUTO
88 . GLOBL S#TWFN, S#TTFN, S#OTFN, S#IOFN, S#OTLO
89 . GLOBL LSTD, FSTD, #DETC, UMSYTP, S#TTSC
90 . GLOBL #DISCN, LPROJ, LPROG, LUNAME, S#RT, S#LOW
91 . GLOBL LCPUI, LCPULO, LCONTM, #CTRLS, #SPLJB, TXTCL
92 . GLOBL STPFLG, TON, USPLCH, SPLCHN, S#HICP
93 . GLOBL S#INWT, S#OTWT, S#TMWT, S#SFWT, S9600
94 . GLOBL S#MSWT, CFBUF, CFEND, CCLSAV, KMNCHN
95 . GLOBL MINTIM, LSECPT, MAXSEC, #EMTTR, VCSHNB
96 . GLOBL OKFILE, OKFEND, #CLTST, UCISPC, MHNSIZ
97 . GLOBL CASTBR, CASCBR, CASTBW, CASCP, MHNSMS
98 . GLOBL CASTRO, CASTWO, CLTOTL, CO#DTR, CLSFSP
99 . GLOBL CO#CR, CO#FF, CO#FFO, CO#LC, CO#TAB, CO#CTL
100 . GLOBL CO#LFI, CO#LFO, CO#BNI, CO#BND, CL#OPT
101 . GLOBL CL#LEN, CL#SKP, CL#WID, CL#LIN, PHYMEM
102 . GLOBL LJSW, CTRLTT, NEWJSW, JSTKND, VIMAGE
103 . GLOBL USTART, #ENTOP, BOTDEV, BOTUNI, CSHALC
104 . GLOBL #CTRLC, LSW2, #INKMN, CHAIN, UFORM
105 . GLOBL #SQO, #SQO3, LITIME
106 . GLOBL MAXASN, #CFABT, INDSTA, INDERR
107 . GLOBL RUNDEV, LNBLKS, CXTBAS, CXTWDS, UHIMEM
108 . GLOBL #DILUP, CSHDEV, CSHDVN, LNSBLK
109 . GLOBL LSW3, LSW2S, #DUPRN
110 . GLOBL #FORM, #TAB, LSCCA, #CFSOT, LOFSPC, R50COM
111 . GLOBL #PAGE, #SCOPE, #ECHO, #LC, #BBIT, CHKALC
112 . GLOBL UCHAN, #FORMO, #CFALL, #CFDCC, #CFCLC
113 . GLOBL LNPRIM, LNMAP, CW#5OH, CONFIG, #SUCF
114 . GLOBL #DOOFF, NUCHN, LRBFIL, CFIND, TALENT

```

115 .GLOBL C. CSW, C. DEVQ, C. SBLK, NLINES, CD$BBT
116 .GLOBL CD$NAM, CD$DVU, CD$BAS, CD$JOB, CD$$SZ, CD$$UB
117 .GLOBL LTSCMD, LNSPAC, CFNEST, UCLNAM
118 .GLOBL $CFOPN, CFSEND, PBFEND, CFSP, $TTGAG
119 .GLOBL UFPTRP, SDSFCB, SD$DEL, CFLFL4, $UCLCF
120 .GLOBL SDFLAG, SD$FLK, SD$WFM, SDFORM, $UCLRN
121 .GLOBL SDBUF1, SDBLK, NSPLDV, LD$RON, $UCLCM, $UCLCL
122 .GLOBL LDNAME, LDSIZE, LDFLAG, LDBASE, LDPDEV
123 .GLOBL LSW8, $SQ01, $SQ01A, $SQ01B, $SQ01C, $SQ02, $SQIID, $SQHIO
124 .GLOBL $DEFER, CFCHAN, SCHAIN, LDDEVX, $SGALL
125 .GLOBL CFPNT, CFBLK, $QUIET, DIABFL
126 .GLOBL DIABNO, VT52NO, LA36NO, LA36FL
127 .GLOBL LSW4, KL4CLR, SDSKIP, SDBU, SD$BAK
128 .GLOBL $INCOR, $KED, VQUNIB, VINTIO, VQUNIC
129 .GLOBL SF$BSY, SFFORM, SD$SNG, SFNMBL, NFRESB
130 .GLOBL SD$HLD, SF$HLD, CURPRM, PRMPNT, SF$1ST
131 .GLOBL LSTPRM, PRMBUF, PRMEND, CFSPND
132 .GLOBL SDFHD, SFFLAG, SFQLNK, CFHOLD, LOGDVU, LOGBAS
133 .GLOBL LCOL, $QTSET, $TECO, CD$TOP, LOGCHK
134 .GLOBL $WILD, ERRSEV, UERSEV, PASLIN, LOGBAS, LOGDVU
135 .GLOBL LSTPL, SDCB, SDCBND, VQUANO, VQUAN3
136 .GLOBL VQUAN1, VQUNIA, VQUAN2, VHIPCT, VQUANO, VQUAN3
137 .GLOBL DCTRD, DCCRD, DCTWR, DCCWR, ASNSRC
138 .GLOBL VCORTM, NUMDCD, VNUMDC, KMPRMT, MXPRMT
139 .GLOBL RDB, RDBEND, RT$NAM, RT$$SZ, CLDEVX, SDDVU
140 .GLOBL SDNAME, SDCBSZ, LSTSL, LSTATE
141 .GLOBL TK1VAL, CINDAT, SYSDAT, SYTIMH, SYTIML
142 .GLOBL BASMAP, LOMAP, HIMAP, JCXPGS
143 .GLOBL SMRSIZ, SRTSIZ, CSHSIZ, TK1SEC
144 .GLOBL TSXLN, TSXSIT, GRT1, TRGRET, LICTXT, SUPCOD, NAMTOP, SUMS, SUCS
145 .GLOBL LPRG1, LPRG2, S$QUSR, S$IOWT, S$SFWT
146 .GLOBL S$SPDB, S$SPCB, SFUSER, SFFILE, VT200, VT2007, VT2008
147 .GLOBL LCBIT, LA36, LA120, VT52, VT100, DIABLO, QUME
148 .GLOBL ADM3A, LTRMTP, LA12FL, LA12NO, VT52FL
149 .GLOBL VT10FL, VT10NO, QUMEFL, QUMENO, ADM3FL
150 .GLOBL VT20FL, VT20NO
151 .GLOBL ADM3NO, SYINDX, SYUNIT, NUMDEV, PNAME
152 .GLOBL OF$DEV, OF$UNT, OF$FIL, OF$FLQ, SYNAME
153 .GLOBL OF$$SZ, OT$RON, RESDEV, $TAPE
154 .GLOBL KMNBAS, ODTBAS, $CTRLD
155 .GLOBL LSW6, $SNWTT, PF$SYS, PF$IOW, $DEBUG
156 .GLOBL RSR, TSR, LMXNUM, LSTMX, MXDTR, ZCLR, MXCSR
157 .GLOBL $INDDF, $INDRN, IN$ACT, IN$CNT, IN$CMD, INDSAV
158 .GLOBL $PHONE, INVEC, LMXLN, MXVEC, $INIT, $DEAD, $HARD
159 .GLOBL ITRMTP, LMXPRM, LSW7, $INDAB, CFSTS, CF$IND, CF$QUT
160 .GLOBL CFABLV, MONVEC, LBSPRI, MAXPRI, MXJPRI, LPRI, $SYSPS
161 .GLOBL LOGCHN, LOGFLQ, LOGPTR, LOGBUF, LOGBLK
162 .GLOBL LF$OPN, LF$WRT, UCLBLK, UCLDAT
163 .GLOBL CSHHD, FC$CDX, FC$LNK, FD$NAM, UC$NDC, UC$MDC, CVTUC
164 .GLOBL CMDBUF, PAUMSQ, RDCMD, DKSAV, SYSAV, CVTTAB, RUNHD, SEARCH
165 .GLOBL INVOPT, FKILL, ABRTCF, ACRFN, XAREA, FILNAM, NOPRG, FPRINT
166 .GLOBL PUSHCF, TRMSTR, FILNAM, R50DIR, R50SY, R50IND, R50SAV
167 .GLOBL INOACT, R50DUP, R50PIP, R50KED, R50K52, R50KEX, R50TSX, R50UCL
168 .GLOBL BLKO, RDERM, R50VIR, NOSTRT, RUNEMT, OVRCOR
169 .GLOBL BADSAV, LDNAM, NOPRG, NOCIN, SIZVAL, ASKLNM, BADCMD, KCSIBF
170 .GLOBL ASDEX, KCSIMS, ASNOVF, QTRD50, R50BUF, R50LDO, MNTDEV, DMTARG
171 .GLOBL DEADEV, CHKMNT, CHKMTX, INFOMT, NOFLAG, MTOPHD, INVOPT, ILLCMD

```

172	. GLOBL	R5OLD, INVLDN, R5ODSK, ACRFIL, BDFNAM, LOGASN, MNTFUL, R5OLD7
173	. GLOBL	TBLQVF, SETHD, CSIMS2, CKPRIV, R5OND, AMBOPT, ACRDEC
174	. GLOBL	MAXAVL, PRTDEC, DEVUNT, PNAME, HNBUF, CKTERM
175	. GLOBL	ACROCT, HANBSY, CSIMS1, MISSEQ, NOIND, POPCF
176	. GLOBL	BADPMT, BADPRI, TOTXT, CRLF, HIPRI, STLQHD, LOGCLS, R5OLOG
177	. GLOBL	BDLQOP, SPLHLA, NOCCL, LDOPHD, PRTFIX, PRTSPC
178	. GLOBL	DLTXT, OCTFIX, PRITTP, NATXT, SPDTX1, NOTXT, YESTXT, NINTXT
179	. GLOBL	PRTUNM, SYHD1, SYHD2, PRTLN, SPACE2, DETTXT, SPACE3, RNMS
180	. GLOBL	SWPTX, LOCKTX, SPACE5, PRTDC3, KBMSG, DIVIDE, PRTDC2
181	. GLOBL	COLOO, CPUAH, CPUAL, PRTTMV, NOFIL, CMDBUF, CALUCL
182	. GLOBL	NOUDC, DEVHD1, ASNHD1, ASNHD2, SHMTH1, SHMTH2, PRTTMD
183	. GLOBL	CVDVNM, SPACE6, PRTBUF, PRTFNM, NONEMS, NODAT, NOLDMT
184	. GLOBL	SUBARO, EDTFIL, RONTXT, NOTAVL, KBTX, MNFLGS, MNBPC
185	. GLOBL	DELSPC, MNBASE, MNTOP, MONHD, MONAR1, NOPMGN, PMBUSY, MONAR2
186	. GLOBL	NSWPMS, MAXMTX, CURMTX, CHKDLM, SPLHD, AMBOPT, INVOPT
187	. GLOBL	DEVIDL, COAL, ALDEX, COAD, SPACTV, SPWFM, DEVIDL, SPSNG
188	. GLOBL	COAL, ALDEX, ALDBLK, COAD, SPACTV, SPWFM, DEVIDL
189	. GLOBL	SPSNG, SPFUL, SPCF, SPFLK, NOFIL, SPGEMT, NOOPTT
190	. GLOBL	BDLIN, MSGBUF, MSGEND, NOTON, GAGMSG
191	. GLOBL	LINFRE, DJABMS, DLMSG, INVTIM, DMTALL
192	. GLOBL	SHTMSG, AUTHFN, SPLACT, DOSTOP, OFFEMT, KILEMT, UPTMMS
193	. GLOBL	TMTOTH, DIVSOR, TMTOTL, PRTPCT, SUM1, SUM2, SUM3, SUM4
194	. GLOBL	SUM5, SUM6, SUM7, OTHRON, SPLPND, STPASK, SRTSMS, CHKTTD
195	. GLOBL	SIZEMT, ASNOVF, INVLDM, CSIMS4, MNTARG, HUPARG, R5OTT
196	. GLOBL	KMNNAM, NOKMON, CCLNAM, OTRMNT, CHKDEV, DMTSUB, CMDCCL
197	. GLOBL	SHOHD, SUBTXT, MNTTXT, SRTTXT, TOTMMS, UMSSMS, SSRMAP
198	. GLOBL	TSXSMS, USRMMS, JCXSMS, DZTXT, OCTPRT
199	. GLOBL	PRTR50, PRTDAT, PRTTOD, PRTTIM, INVDEV, ALFN, R5ODK
200	. GLOBL	DETHD, DETARG, RUNMS, NOFRDL, R5OMON, INV DAT, MUL32, COAF
201	. GLOBL	AR\$PRJ, AR\$PRG, AR\$CON, AR\$CNT, AR\$CPH, AR\$CPL, AR\$UNM
202	. GLOBL	AR\$DMY, AR\$\$SZ, ARNRPB, \$SLON, \$SLTTY, \$SLLET
203	. GLOBL	PRTWRN, SLMXLN, VSLEDT, \$LOFCF, CSHMSG
204	. GLOBL	AF\$HIE, AF\$NOI, \$NOINT, AF\$PLK, AF\$DBG
205	. GLOBL	AF\$IOP, \$RNIOP

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

```

1      ; -----
2      ; Macro to cause a fatal error message to be printed.
3      ;
4      .MACRO FERR MSG
5      MOV R5, -(SP)
6      MOV MSG, R5
7      CALL FPRINT
8      MOV (SP)+, R5
9      .ENDM FERR
10     ;
11     ; -----
12     ; Macro to print a fatal error message, clean up
13     ; and then jump to RDCMD.
14     ;
15     .MACRO FABORT MSG
16     MOV MSG, R5
17     JMP FKILL
18     .ENDM FABORT
19     ;
20     ; -----
21     ; Macro to print a warning message
22     ;
23     .MACRO FWARN MSG
24     MOV R5, -(SP)
25     MOV MSG, R5
26     CALL PRTWRN
27     MOV (SP)+, R5
28     .ENDM FWARN
29     ;
30     ; -----
31     ; Macro to start a standard option table.
32     ; Name = 1 to 4 character table name.
33     ; NA = Number of arguments per table entry.
34     ;
35     .MACRO TBLDEF NAME, NA
36     NARGS = NA
37     .CSECT CMDVS2
38     NAME 'HD: .WORD 2*NA
39     .ENDM TBLDEF
40     ;
41     ; -----
42     ; Macro to enter an option text name and a set of parameters
43     ; into the currently open table.
44     ; STRNG = Ascii name
45     ; A,B,C = Set of option parameters to store in table with name.
46     ;
47     .MACRO CMDDEF STRNG, A, B, C
48     .CSECT NAMES2
49     L =
50     .ASCIZ /STRNG/
51     .CSECT CMDVS2
52     .WORD L ; POINTER TO NAME STRING
53     .WORD A
54     .IIF GE, <NARGS-2> .WORD B
55     .IIF GE, <NARGS-3> .WORD C
56     .ENDM CMDDEF
57     ;

```

58
59
60
61
62
63
64
65

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

```

1          ; -----
2          ; Data areas
3          ;
4 000000 125017 074773 R50HST: .RAD50 /#HOST#/
5 000004 012256 000000 000000 CLDEV: .RAD50 /CLN /
   000012 000000
6 000014 012276 R50CLO: .RAD50 /CLO/
7 000016 013666 R50C10: .RAD50 /C10/
8 000020 000000 CPFLAG: .WORD 0
9          ;
10         ; Flags stored in CPFLAG
11         ;
12         000001 CPFSUS = 1 ;Suspend process
13         000002 CPFRES = 2 ;Resume process
14         000004 CPFAUT = 4 ;Set authorized privileges
15         ;
16 000022 000 SETPRM: .BYTE 0 ;Indicates if SET is temp or perm
17 000023 000 CPPID: .BYTE 0 ;Process ID for SET PROCESS command
18 000024 000 CPPRIQ: .BYTE 0 ;Priority value for SET PROCESS command
19 000025 CPNAME: .BLKB 12 ;Name buffer for SET PROCESS command
   .EVEN
20         ;
21         ;
22         ; Emt to assign a CL line to a time-sharing line
23         ;
24 000042 000 155 CLAEMT: .BYTE 0,155
25 000044 000000 .WORD 0 ;CL unit number
26 000046 000000 .WORD 0 ;Line number
27         ;
28         ; Emt to cross connect our time-sharing line with a CL unit
29         ;
30 000050 000 126 TTXCL: .BYTE 0,126
31 000052 000013 .WORD 13
32 000054 000000 .WORD 0
33         ;
34         ; Emt to set line speed
35         ;
36 000056 000 154 LSPEMT: .BYTE 0,154
37 000060 000000 .WORD 0 ;Line number
38 000062 000000 .WORD 0 ;Speed code
39         ;
40         ; Emt to reset XOFF status for a line
41         ;
42 000064 001 154 XONEMT: .BYTE 1,154
43 000066 000000 .WORD 0
44         ;
45         ; EMT to set/reset DTR for a line
46         ;
47 000070 002 154 DTREMT: .BYTE 2,154 ;Assume raise DTR (3,154 = drop DTR)
48 000072 000000 .WORD 0
49         ;
50         ; Emt to set privilege flags
51         ;
52 000074 001 150 PVSEMT: .BYTE 1,150
53 000076 002 001 .BYTE 2,1
54 000100 0000000 .WORD PFSO
55 000102 000000 .WORD 0
56         ;

```

```
57          ; Emt to clear privilege flags
58          ;
59 000104      001      150      PVCEMT: .BYTE 1,150
60 000106      001      001          .BYTE 1,1
61 000110      0000000          .WORD PFCO
62 000112      000000          .WORD 0
63          ;
64          ; Emt to start an inactive line
65          ;
66 000114      000      126      STAEMT: .BYTE 0,126
67 000116      000020          .WORD 20
68 000120      000000          .WORD 0
69          ;
70          ; Emt to declare terminal type change to process windowing system
71          ;
72 000122      007      161      WINSTT: .BYTE 7,161
```

```

1
2 ; -----
3 ; Options for SET TT command
4 ;
5 TBLDEF TT,3
6 CMDDEF ADM*3A, SADM3A, 0, 0
7 CMDDEF AUTO*BAUD, SAUTO, 0, 0
8 CMDDEF NOAUTO*BAUD, CLRRTTP, LSW2, $AUTO
9 CMDDEF BIT*S, STCLEN, 0, 0
10 CMDDEF DEAD, SETDED, LSW3, $DEAD
11 CMDDEF NODEAD, CLRRTTP, LSW3, $DEAD
12 CMDDEF DEC*WRITER, STLA36, 0, 0
13 CMDDEF DEF*ER, SETTTB, LSW2, $DEFER
14 CMDDEF NOD*EFER, CLRRTTB, LSW2, $DEFER
15 CMDDEF DI*ABLD, SDIAB, 0, 0
16 CMDDEF DT*R, SETDTR, 0, 0
17 CMDDEF NODT*R, CLRDTR, 0, 0
18 CMDDEF E*CHO, SETTTB, LSW2, $ECHO
19 CMDDEF NOE*CHO, CLRRTTB, LSW2, $ECHO
20 CMDDEF EIGHT*BIT, SETTTB, LSW2, $8BIT
21 CMDDEF NOEIGHT*BIT, CLRRTTB, LSW2, $8BIT
22 CMDDEF FORM, SETTTB, LSW2, $FORM
23 CMDDEF NOFORM, CLRRTTB, LSW2, $FORM
24 CMDDEF FORM0, SETTTB, LSW4, $FORM0
25 CMDDEF NOFORM0, CLRRTTB, LSW4, $FORM0
26 CMDDEF G*AG, SETTTB, LSW7, $TTGAG
27 CMDDEF NOG*AG, CLRRTTB, LSW7, $TTGAG
28 CMDDEF HAZEL*TINE, STHAZL, 0, 0
29 CMDDEF HO*LD, RDCMD, 0, 0
30 CMDDEF NOH*GLD, RDCMD, 0, 0
31 CMDDEF LA1*20, SLA120, 0, 0
32 CMDDEF LA3*6, STLA36, 0, 0
33 CMDDEF LC, SETTTB, LSW2, $LC
34 CMDDEF NOL*C, CLRRTTB, LSW2, $LC
35 CMDDEF LE*NGTH, RDCMD, 0, 0
36 CMDDEF PAG*E, SETTTB, LSW2, $PAGE
37 CMDDEF NOP*AGE, CLRRTTB, LSW2, $PAGE
38 CMDDEF PAR*ITY, SETPAR, 0, 0
39 CMDDEF NOPAR*ITY, STNOPR, 0, 0
40 CMDDEF PHO*NE, SETTTP, LSW2, $PHONE
41 CMDDEF NOPHO*NE, CLRRTTP, LSW2, $PHONE
42 CMDDEF QUI*ET, SETQUT, 0, 0
43 CMDDEF NOQ*UIET, RSTQUT, 0, 0
44 CMDDEF QUM*E, SQUME, 0, 0
45 CMDDEF SC*OPE, SETTTB, LSW2, $SCOPE
46 CMDDEF NOSC*OPE, SETNSC, 0, 0
47 CMDDEF SEVEN*BIT, CLRRTTB, LSW2, $8BIT
48 CMDDEF SI*NGLE, SETTTB, LSW6, $STSNG
49 CMDDEF NOSI*NGLE, CLRRTTB, LSW6, $STSNG
50 CMDDEF SPE*ED, STTTSP, 0, 0
51 CMDDEF STA*RT, STTTST, 0, 0
52 CMDDEF SYSP*ASSWORD, SETTTP, LSW2, $SYSPS
53 CMDDEF NOSYSP*ASSWORD, CLRRTTP, LSW2, $SYSPS
54 CMDDEF SYSPS, SETTTP, LSW2, $SYSPS
55 CMDDEF NOSYSPS, CLRRTTP, LSW2, $SYSPS
56 CMDDEF SYSP*WD, SETTTP, LSW2, $SYSPS
57 CMDDEF NOSYSP*WD, CLRRTTP, LSW2, $SYSPS
58 CMDDEF TAB, SETTTB, LSW2, $TAB

```

```

58 000652          CMDDEF  NOTAB, CLRTTB, LSW2, $TAB
59 000662          CMDDEF  TAP*E, SETTTB, LSW2, $TAPE
60 000672          CMDDEF  NOTAP*E, CLRTTB, LSW2, $TAPE
61 000702          CMDDEF  TR*ANSLATE, STTRNS, 0, 0
62 000712          CMDDEF  VT1*00, SVT100, 24, , 0
63 000722          CMDDEF  VT2*00, SVT200, 24, , 0
64 000732          CMDDEF  VT220, SVT200, 24, , 0
65 000742          CMDDEF  VT240, SVT200, 24, , 0
66 000752          CMDDEF  VT241, SVT200, 24, , 0
67 000762          CMDDEF  VT5*0, SVT50, 12, , 0
68 000772          CMDDEF  VT5*2, SVT50, 24, , 0
69 001002          CMDDEF  W*AIT, SETTW, 0, 0
70 001012          CMDDEF  NOW*AIT, SETTNW, 0, 0
71 001022          CMDDEF  XON, SEIXON, 0, 0
72 001032          CMDDEF  8BIT, SETTTB, LSW2, $8BIT
73 001042          CMDDEF  NO8BIT, CLRTTB, LSW2, $8BIT
74 001052          CMDDEF  7BIT, CLRTTB, LSW2, $8BIT
75 001062          TBLEND
76
77          ; Define options for SET Tl [n] PARITY= command
78          ;
79 000124          TBLDEF  PAR, 3
80 001066          CMDDEF  E*VEN, STPAR, LP$PAR
81 001076          CMDDEF  O*DD, STPAR, LP$PAR!LP$ODD
82 001106          CMDDEF  N*ONE, STPAR, 0
83 001116          TBLEND
84
85          ;
86          ; Define options for the SET CLn command
87          ;
88 000124          TBLDEF  CLOP, 2
89 001122          CMDDEF  BIT*S, STCLB!, 0
90 001130          CMDDEF  CR, STCLOP, CO$CR
91 001136          CMDDEF  NOCR, CLCLOP, CO$CR
92 001144          CMDDEF  DTR, STCLOP, CO$DTR
93 001152          CMDDEF  NODTR, CLCLOP, CO$DTR
94 001160          CMDDEF  ENDP*AGES, STCLVL, CL$EPN
95 001166          CMDDEF  ENDS*TRING, CLESTR, 0
96 001174          CMDDEF  FORM, STCLOP, CO$FF
97 001202          CMDDEF  NOFORM, CLCLOP, CO$FF
98 001210          CMDDEF  FF, STCLOP, CO$FF
99 001216          CMDDEF  NOFF, CLCLOP, CO$FF
100 001224         CMDDEF  FORMO, STCLOP, CO$FFO
101 001232         CMDDEF  NOFORMO, CLCLOP, CO$FFO
102 001240         CMDDEF  GR*APH, STCLOR, 0.
103 001246         CMDDEF  NOGR*APH, STCLOR, 132.
104 001254         CMDDEF  LC, STCLOP, CO$LC
105 001262         CMDDEF  NOLC, CLCLOP, CO$LC
106 001270         CMDDEF  TAB, STCLOP, CO$TAB
107 001276         CMDDEF  NOTAB, CLCLOP, CO$TAB
108 001304         CMDDEF  CTRL, STCLOP, CO$CTL
109 001312         CMDDEF  NOCTRL, CLCLOP, CO$CTL
110 001320         CMDDEF  LEN*$TH, STCLVL, CL$LEN
111 001326         CMDDEF  LFIN, STCLOP, CO$LFI
112 001334         CMDDEF  NOLFIN, CLCLOP, CO$LFI
113 001342         CMDDEF  LFOUT, STCLOP, CO$LFO
114 001350         CMDDEF  NOLFOUT, CLCLOP, CO$LFO

```

```

115 001356      CMDDEF  LIN*E, STCLLN, 0
116 001364      CMDDEF  BININ, STCLOP, CO*BNI
117 001372      CMDDEF  NOBININ, CLCLOP, CO*BNI
118 001400      CMDDEF  BINOUT, STCLOP, CO*BNO
119 001406      CMDDEF  NOBINOUT, CLCLOP, CO*BNO
120 001414      CMDDEF  8BIT, STCLOP, CO*8BT
121 001422      CMDDEF  EIGHT*BIT, STCLOP, CO*8BT
122 001430      CMDDEF  NO8*BIT, CLCLOP, CO*8BT
123 001436      CMDDEF  NOEIGHT*BIT, CLCLOP, CO*8BT
124 001444      CMDDEF  7BIT, CLCLOP, CO*8BT
125 001452      CMDDEF  PAR*ITY, STCLPD, CLPRHD
126 001460      CMDDEF  NOPAR*ITY, STCLNP
127 001466      CMDDEF  SEVEN*BIT, CLCLOP, CO*8BT
128 001474      CMDDEF  TOP, CLTOP, 0
129 001502      CMDDEF  TR*ANSLATE, CLTRNS, 0
130 001510      CMDDEF  RES*ET, STCLRS, 0
131 001516      CMDDEF  SKI*P, STCLVL, CL$SKP
132 001524      CMDDEF  WID*TH, STCLVL, CL$WID
133 001532      CMDDEF  SP*EED, STCLSP, 0
134 001540      CMDDEF  VER*SION, STCLVR, 0
135 001546      CMDDEF  XON, STCLXN, 0
136 001554      TBLEND
137
;
138      ; Define options for SET CL PARITY= command
139      ;
140      TBLDEF  CLPR, 2
141      CMDDEF  E*VEN, STCLPR, LP$PAR
142      CMDDEF  O*DD, STCLPR, LP$PAR!LP$ODD
143      CMDDEF  N*ONE, STCLPR, 0
144      TBLEND
145
;
146      ; Define options for the SET HOST command
147      ;
148      TBLDEF  HOST, 1
149      CMDDEF  DTE, HSTDTE
150      CMDDEF  CL, HSTDTE
151      CMDDEF  PO*RT, HSTPR1
152      TBLEND
153
;
154      ; Define options for SET PROCESS command
155      ;
156      TBLDEF  CPOP, 2
157      CMDDEF  PRIV*ILEGES, PRVOPT, 0
158      CMDDEF  ID*ENTIFICATION, CPPPID, 0
159      CMDDEF  NAM*E, CPPNAM, 0
160      CMDDEF  PRIO*RITY, CPPPRI, 0
161      CMDDEF  SUSP*END, CPPFLG, CPF SUS, 0
162      CMDDEF  RES*UME, CPPFLG, CPFRES, 0
163      CMDDEF  AU*THORIZED, CPPFLG, CPF AUT, 0
164      TBLEND

```

SET commands

```

1          .SBTTL  SET commands
2          .SBTTL  .      Process
3          ;-----
4          ; Process the SET PROCESS command.
5          ;
6 000124   SETPRC:
7          ;
8          ; Initialize cells which will hold values parsed by the command
9          ;
10 000124  116767  0000000 177671      MOVB   CORUSR, CPPID      ;Set process ID=our job
11 000132  105067  177667              CLRB   CPNAME          ;No name specified
12 000136  105067  177662              CLRB   CPPRID         ;No priority
13 000142  005067  177652              CLR    CPFLAG        ;No flags
14 000146  004767  0000000            CALL   CLRPRV         ;No privileges
15          ;
16          ; Do command parsing
17          ;
18 000152  012704  001624'            MOV    #CPOPHD, R4     ;Point to option driver list
19 000156  004767  0000000            CALL   OPTLST         ;Parse the command
20          ;
21          ; Now process each option that was accrued
22          ;
23 000162  004767  000366              CALL   CPSNAM         ;See if process name was specified
24 000166  004767  000644              CALL   CPSPRI        ;See if priority was specified
25 000172  004767  000464              CALL   CPSPRV        ;See if privileges were specified
26 000176  004767  000676              CALL   CPSFLG        ;See if flags were specified
27          ;
28          ; Finished command
29          ;
30 000202  000167  0000000          9#:   JMP    RDCMD      ;Finished command

```

Process

```

1          ; -----
2          ; SET PROCESS/IDENTIFICATION=job-number
3          ;
4 000206 010146 CPPPID: MOV      R1, -(SP)
5          ;
6          ; Accrue the ID value
7          ;
8 000210 004767 0000000 CALL   ACRDEC
9          ;
10         ; Make sure the ID value is valid
11        ;
12 000214 006301          ASL     R1          ; Convert to word table index
13 000216 001413          BEQ     10$         ; Invalid if zero
14 000220 020127 0000000 CMP     R1, #LSTSL      ; Is this a valid job #?
15 000224 101010          BHI     10$         ; Br if not
16 000226 032761 0000000 0000000 BIT     #$DILUP, LSW(R1) ; Is line logged on?
17 000234 001410          BEQ     11$         ; Br if not
18 000236 110167 177561          MOVB   R1, CPPPID      ; Save ID value
19        ;
20        ; Finished
21        ;
22 000242 012601          MOV     (SP)+, R1
23 000244 000207          RETURN
24        ;
25        ; Invalid ID
26        ;
27 000246          10$:   FABORT  #EM$ILN      ; Invalid job numver
28 000256          11$:   FABORT  #NOTON      ; Line not logged on

```

Process

```

1          ; -----
2          ;   SET PROCESS/PRIORITY=value
3          ;
4 000266   010146 CPPPRI: MOV     R1,-(SP)
5 000270   010546         MOV     R5,-(SP)
6          ;
7          ;   Accrue the priority value
8          ;
9 000272   004767   0000000          CALL   ACRDEC           ;Accrue priority value
10         ;
11        ;   See if the priority value is valid
12        ;
13 000276   005701          TST     R1           ;Must be > 0
14 000300   003406          BLE     10$           ;Br if invalid
15 000302   120167   0000000          CMPB   R1,MXJPRI       ;Is it within valid range for job?
16 000306   101003          BHI     10$           ;Br if not
17 000310   110167   177510          MOVB  R1,CPPRI0       ;Save priority value
18 000314   000424          BR     9$
19        ;
20        ;   Invalid priority value
21        ;
22 000316          10$:  FERR   #BADPRI       ;Invalid priority value
23 000332   012705   0000001          MOV   #1,R5           ;Minimum prio value
24 000336   004767   0000000          CALL  PRTDEC
25 000342          .PRINT #TOTXT           ;"to"
26 000350   116705   0000000          MOVB  MXJPRI,R5       ;Max possible
27 000354   004767   0000000          CALL  PRTDEC
28 000360          .PRINT #CRLF
29        ;
30        ;   Finished
31        ;
32 000366   012605          9$:  MOV   (SP)+,R5
33 000370   012601          MOV   (SP)+,R1
34 000372   000207          RETURN

```

Process

```

1
2 ; -----
3 ; SET PROCESS/NAME="string"
4 000374 010246 CPPNAM: MOV R2, -(SP)
5 000376 010446 MOV R4, -(SP)
6
7 ; Make sure equal sign follows keyword
8 ;
9 000400 004767 0000000 CALL CHKEQ ; Check for equal sign
10 000404 012702 000025' MOV #CPNAME, R2 ; Point to buffer for name
11
12 ; See if string is enclosed in quote marks
13 ;
14 000410 004767 0000000 CALL SKPSPC ; Skip over any spaces
15 000414 111300 MOV (R3), R0 ; Get 1st char of string
16 000416 120027 000042 CMPB R0, #42 ; Quote mark?
17 000422 001403 BEQ 3$ ; Br if yes
18 000424 120027 000047 CMPB R0, #47 ; Apostrophe?
19 000430 001014 BNE 1$ ; Br if not
20
21 ; String is enclosed in quote marks
22 ;
23 000432 004767 0000000 3$: CALL ACRSTR ; Accrue a quoted string
24 000436 005700 TST R0 ; Is string empty?
25 000440 001423 BEQ 2$ ; If empty, go blank fill it.
26 000442 020027 000014 CMP R0, #12. ; Is string too long?
27 000446 101026 BHI 10$ ; Br if too long
28 000450 012704 0000000 MOV #BLKO, R4 ; Point to accrued string
29 000454 112422 4$: MOV (R4)+, (R2)+ ; Move to string buffer
30 000456 077002 SOB R0, 4$
31 000460 000413 BR 2$ ; Go blank fill rest of buffer if needed
32
33 ; Name string is not quoted.
34 ; Move name to buffer till we hit end of string
35 ;
36 000462 111300 1$: MOV (R3), R0 ; Get next char from string
37 000464 001411 BEQ 2$ ; Br if hit end of command
38 000466 120027 000057 CMPB R0, #'/' ; Start of next qualifier?
39 000472 001406 BEQ 2$ ; Br if yes
40 000474 005203 INC R3 ; Point to next character in command
41 000476 020227 000041' CMP R2, #CPNAME+12. ; Buffer overflow?
42 000502 103010 BHIS 10$ ; Br if yes
43 000504 110022 MOV R0, (R2)+ ; Store char into name buffer
44 000506 000765 BR 1$
45
46 ; Fill remainder of buffer with blanks
47 ;
48 000510 020227 000041' 2$: CMP R2, #CPNAME+12. ; Filled buffer yet?
49 000514 103011 BHIS 9$ ; Br if yes
50 000516 112722 000040 MOV #' ', (R2)+ ; Pad with blanks
51 000522 000772 BR 2$
52
53 ; Error: string is too long
54 ;
55 000524 10$: FERR #EM$STL ; String is too long
56
57 ; Finished

```

Process

```
58 ;
59 000540 012604 9*: MOV (SP)+,R1
60 000542 012602 MOV (SP)+,R2
61 000544 000207 RETURN
62 ;-----
63 ; Set some flag for a SET PROCESS qualifier
64 ;
65 000546 051467 177246 CPPFL0: BIS (R4),CPFLAG ;Set specified flag bit
66 000552 000207 RETURN
```

Process

```

1 ;-----
2 ; See if NAME qualifier was specified.
3 ;
4 000554 010146 CPSNAM: MOV R1, -(SP)
5 000556 010246 MOV R2, -(SP)
6 ;
7 ; See if name qualifier was specified
8 ;
9 000560 105767 177241 TSTB CPNAME ;Was name qualifier specified?
10 000564 001423 BEQ 9$ ;Br if not
11 ;
12 ; Make sure we are changing for our own process
13 ;
14 000566 116701 177231 MOVB CPPID, R1 ;Get job index
15 000572 120167 0000000 CMPB R1, CORUSR ;Affecting our own process?
16 000576 001021 BNE 10$ ;Br if not
17 000600 032767 0000000 0000000 BIT #PO$NAM, PRIVCO ;Are we authorized to change our name?
18 000606 001421 BEQ 11$ ;Br if not
19 000610 070127 000006 MUL #6, R1 ;Each job has 12 character name
20 000614 062701 0000000 ADD #LUNAME, R1 ;Point to name cell for our job
21 000620 012702 000025' MOV #CPNAME, R2 ;Point to name buffer
22 000624 012700 000014 MOV #12, R0 ;Get # bytes to move
23 000630 112221 1$: MOVB (R2)+, (R1)+ ;Set name for job
24 000632 077002 SOB R0, 1$
25 ;
26 ; Finished
27 ;
28 000634 012602 9$: MOV (SP)+, R2
29 000636 012601 MOV (SP)+, R1
30 000640 000207 RETURN
31 ;
32 ; Cannot change name of another job
33 ;
34 000642 10$: FABORT #EM$CNO
35 000652 11$: FABORT #EM$NPR

```

```

1          ; -----
2          ;   Change privilege flags for our process.
3          ;
4 000662 010246 CPSPRV: MOV      R2, -(SP)
5 000664 010346          MOV      R3, -(SP)
6 000666 010446          MOV      R4, -(SP)
7          ;
8          ;   See if any privilege changes were specified
9          ;
10 000670 012702 0000000 MOV      #PFS0, R2      ;Flags to set
11 000674 012703 0000000 MOV      #PFC0, R3      ;Flags to clear
12 000700 012700 0000000 MOV      #PVNPW, R0     ;# words to check
13 000704 005722 1#: TST      (R2)+    ;Any flags to set?
14 000706 001004          BNE      2$      ;Br if yes
15 000710 005723          TST      (R3)+    ;Any flags to clear?
16 000712 001002          BNE      2$      ;Br if yes
17 000714 077005          SOB      R0, 1$      ;Keep checking
18 000716 000433          BR       9$      ;No privilege changes requested
19          ;
20          ;   Make sure we are changing privileges for our job
21          ;
22 000720 126767 177077 0000000 2#: CMPB     CPPID, CORUSR   ;Changing privilege for our job?
23 000726 001033          BNE      10$      ;Br if not
24          ;
25          ;   Set correct EMT function code depending on whether we are doing a
26          ;   normal privilege change or are changing authorized privileges.
27          ;
28 000730 012704 0000001          MOV      #1, R4      ;Assume normal priv change
29 000734 032767 0000004 177056 BIT      #CPFAUT, CPFLAG ;Changing authorized privileges?
30 000742 001402          BEQ      4$      ;Br if not
31 000744 012704 0000002          MOV      #2, R4      ;Code to change authorized privileges
32 000750 110467 177123 4#: MOV      R4, PVSEMT+3 ;Set code in EMT arg block
33 000754 110467 177127          MOV      R4, PVCEMT+3
34          ;
35          ;   Set any specified privileges
36          ;
37 000760 005004          CLR      R4      ;No privilege error detected
38 000762 012700 0000074' MOV      #PVSEMT, R0    ;Point to EMT arg block
39 000766 104375          EMT      375        ;Set privilege flags
40 000770 103001          BCC     3$      ;Br if no authorization violation
41 000772 005204          INC      R4      ;Remember we had authorization violation
42          ;
43          ;   Clear any specified privileges
44          ;
45 000774 012700 000104' 3#: MOV      #PVCEMT, R0 ;Point to EMT arg block
46 001000 104375          EMT      375        ;Clear privilege flags
47 001002 005704          TST      R4      ;Did we have an authorization violation?
48 001004 001010          BNE     11$      ;Br if yes
49          ;
50          ;   Finished
51          ;
52 001006 012604 9#: MOV      (SP)+, R4
53 001010 012603          MOV      (SP)+, R3
54 001012 012602          MOV      (SP)+, R2
55 001014 000207          RETURN
56          ;
57          ;   Can't change privilege for another job
    
```

58
59 001016
60
61
62
63 001026

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

```
1 ;-----  
2 ; See if we need to change priority for job.  
3 ;  
4 001036 010246 CPSPRI: MOV R2, -(SP)  
5 ;  
6 ; See if a priority change was requested  
7 ;  
8 001040 105767 176760 TSTB CPPRIO ;Was a priority specified  
9 001044 001413 BEQ 9$ ;Br if not  
10 001046 116702 176751 MOVB CPPID, R2 ;Get ID # of job we want to affect  
11 001052 004767 0000000 CALL CKACDJ ;Can we access that job?  
12 001056 103406 BCS 9$ ;Br if not  
13 ;  
14 ; Change priority value  
15 ;  
16 001060 116762 176740 0000000 MOVB CPPRIO, LBPRI(R2) ;Set base priority  
17 001066 116762 176732 0000000 MOVB CPPRIO, LPRI(R2) ;Set running priority  
18 ;  
19 ; Finished  
20 ;  
21 001074 012602 9$: MOV (SP)+, R2  
22 001076 000207 RETURN
```

Process

```

1 ; -----
2 ; See if we need to suspend or resume a job.
3 ;
4 001100 010246 CPSFL0: MOV R2, -(SP)
5 ;
6 ; See if any flags were specified
7 ;
8 001102 032767 000003 176710 BIT #<CPFSUS!CPFRES>, CPFLAG ; Suspend or resume requested?
9 001110 001423 BEQ 9$ ; Br if not
10 ;
11 ; Make sure we aren't suspending or resuming our own job
12 ;
13 001112 116702 176705 MOV# CPPID, R2 ; Get ID # of job we want to affect
14 001116 120267 000000# CMPB R2, CDRUSR ; Is it our job?
15 001122 001420 BEQ 10$ ; Br if yes
16 ;
17 ; Make sure we can access the specified job
18 ;
19 001124 004767 000000# CALL CKACD#J ; Can we access that job?
20 001130 103413 BCS 9$ ; Br if not
21 ;
22 ; Suspend or resume the specified job
23 ;
24 001132 012700 000000# MOV #RJEMT, R0 ; Point to resume-job arg block
25 001136 032767 000002 176654 BIT #CPFRES, CPFLAG ; Does he want to resume the job?
26 001144 001002 BNE 1$ ; Br if yes
27 001146 012700 000000# MOV #SJEMT, R0 ; Point to suspend-job arg block
28 001152 010260 000004 1$: MOV R2, 4(R0) ; Set job # in EMT arg block
29 001156 104375 EMT 375 ; Suspend or resume the job
30 ;
31 ; Finished
32 ;
33 001160 012602 9$: MOV (SP)+, R2
34 001162 000207 RETURN
35 ;
36 ; Error -- Trying to suspend or resume our own job
37 ;
38 001164 10$: FABORT #EM$IDR ; /ID parameter required

```

Terminal

```

1
2
3
4
5 001174
6
7
8
9 001174 105067 176622
10 001200 004767 0000000
11 001204 010305
12 001206 005002
13 001210 112500
14 001212 001413
15 001214 120027 000040
16 001220 001410
17 001222 120027 000060
18 001226 103435
19 001230 120027 000071
20 001234 101032
21 001236 005202
22 001240 000763
23 001242 005702
24 001244 001426
25
26
27
28 001246 004767 0000000
29
30
31
32 001252 006301
33 001254 001403
34 001256 020127 0000000
35 001262 101404
36 001264
37
38
39
40 001274 004767 0000000
41 001300 110167 176516
42
43
44
45
46
47 001304 032761 0000000 0000000
48 001312 001003
49 001314 016161 0000000 0000000
50
51
52
53
54
55 001322 012704 0000000
56 001326 004767 0000000
57

```

```

.SBTTL Terminal
-----
; Process the SET TERMINAL command.
;
SETTTY:
;
; See if a line number was specified
;
          CLRB   SETPRM      ; Assume no line number specified
          CALL   SKPSPC      ; Skip over any spaces
          MOV    R3,R5       ; Save pointer to first operand
          CLR    R2          ; Say no digits seen yet
1$:      MOVB   (R5)+,R0     ; Get next char
          BEQ    2$          ; Br if hit end of command line
          CMPB   R0,#40      ; Is this a space?
          BEQ    2$          ; Br if yes
          CMPB   R0,#'0      ; Is this a digit?
          BLO   3$          ; Br if not
          CMPB   R0,#'9      ; Br if not digit
          INC    R2          ; Remember we saw a digit
          BR     1$          ; Scan first item and see if it is a number
2$:      TST    R2           ; Was first item a number?
          BEQ    3$          ; Br if not
;
; There is a terminal number specified, accrue it
;
          CALL   ACRDEC      ; Accrue the terminal number
;
; See if this is a valid terminal number
;
          ASL    R1          ; Convert line number to line index number
          BEQ    4$          ; Zero is not valid line number
          CMP    R1,#LSTPL   ; Is this a primary line number?
          BLOS   5$          ; Br if yes
4$:      FABORT #EM$ILN     ; Invalid line number
;
; Require TERMINAL privilege to make any permanent change
;
5$:      CALL   CKTERM      ; Require TERMINAL privilege to do this
          MOVB   R1,SETPRM   ; Remember to make set permanent
;
; If we are changing line parameters for a line that is not
; currently logged on, initialize the LSW2 table from the initial
; flags for the line (ILSW2).
;
          BIT    #$DILUP,LSW(R1) ; Is line logged on?
          BNE    3$          ; Br if yes
          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.
;
3$:      MOV    #TTHD,R4     ; Get pointer to option table
          CALL   SCNOPS      ; Process all of the command options
;

```

58 ; End of command reached
59 ;
60 001332 000167 0000000 JMP RDCMD ;Finished SET TT command

Terminal

```

1 ;-----
2 ; SET TERMINAL n START
3 ;
4 001336 004767 0000000 STTTST: CALL CKPRIV ;Require OPER privilege
5 ;
6 ; See if line is already started
7 ;
8 001342 032761 0000000 0000000 BIT #DILUP,LSW(R1) ;Is line already started?
9 001350 001404 BEQ 1$ ;Br if not
10 001352 FABORT #EM$LAS ;Line is already started
11 ;
12 ; See if line is marked dead
13 ;
14 001362 032761 0000000 0000000 1$: BIT #DEAD,LSW3(R1) ;Is line marked as dead?
15 001370 001404 BEQ 2$ ;Br if not
16 001372 FABORT #EM$LFD ;Line is dead
17 ;
18 ; See if a CL unit is attached to the line
19 ;
20 001402 005761 0000000 2$: TST LCLUNT(R1) ;Is line being used by a CL unit?
21 001406 002404 BLT 3$ ;Br if not
22 001410 FABORT #EM$CLU ;Line in use by a CL unit
23 ;
24 ; Start the line
25 ;
26 001420 010167 176474 3$: MOV R1,STAEMT+4 ;Store line index into EMT arg block
27 001424 012700 000114' MOV #STAEMT,R0 ;Point to EMT arg block
28 001430 104375 EMT 375 ;Start the line
29 001432 000207 9$: RETURN

```

Terminal

```

1          ;-----
2          ; SET TERMINAL [n] SPEED=n
3          ;
4 001434 004767 0000000 STTTSP: CALL CKTERM          ;Require TERMINAL privilege
5 001440 004767 0000000          CALL ACRSPD          ;Accrue speed value
6 001444 042761 0000000 0000000          BIC  ##AUTO,LSW2(R1) ;Say no autobaud on this line
7 001452 116100 0000010          MOV  LMXPRM+1(R1),R0 ;Get control flags for line
8 001456 042700 0000000          BIC  #<LP#SPD>,R0 ;Clear speed flags
9 001462 050000          BIS  R0,R5 ;Combine parity and length flags with speed
10 001464 010567 176372 STSP1:  MOV  R5,LSPEMT+4 ;Store speed, par, len code into EMT arg block
11 001470 010167 176364          MOV  R1,LSPEMT+2 ;Store line # into EMT arg block
12 001474 006267 176360          ASR  LSPEMT+2 ;Convert index # to line #
13 001500 012700 000056'          MOV  #LSPEMT,R0 ;Point to EMT arg block
14 001504 104375          ENT  375 ;Set the speed
15 001506 103402          BCS  2# ;Br if error
16 001510 000167 000474          JMP  BSO1
17          ;
18          ; Error on EMT to set speed
19          ;
20 001514 2#: FABORT #EM#ILN ;Say invalid line number

```

Terminal

```

1
2 ; -----
3 ; SET TT [n] XON
4 001524 004767 0000000 SETXON: CALL CKTERM ;Require TERMINAL privilege
5 001530 010167 176332 MOV R1,XONEMT+2 ;Set line # in EMT argument block
6 001534 006267 176326 ASR XONEMT+2 ;Convert index # to line #
7 001540 012700 000064' MOV #XONEMT,R0 ;Point to EMT argument block
8 001544 104375 EMT 375 ;Do the XON EMT
9 001546 103402 BCS 2# ;Br if invalid line #
10 001550 000167 000434 JMP BSO1
11 001554 2#: FABORT #EM$ILN ;Say invalid line number

```

Terminal

```

1
2 ; -----
3 ; SET TT [n] [NO]DTR
4 001564 112767 000002 176276 SETDTR: MOVB #2,DTREMT ;Set function to raise DTR
5 001572 000403 BR DTRCOM ;Br to common routine
6 001574 112767 000003 176266 CLRDR: MOVB #3,DTREMT ;Set function to lower DTR
7 001602 004767 0000000 DTRCOM: CALL CKTERM ;Require TERMINAL privilege
8 001606 006201 ASR R1 ;Convert line index to line number
9 001610 010167 176256 MOV R1,DTREMT+2 ;Set in EMT arg block
10 001614 006301 ASL R1 ;Get back as line index
11 001616 012700 000070' MOV #DTREMT,R0 ;Point to EMT arg block to
12 001622 104375 EMT 375 ;Raise or lower DTR
13 001624 103402 BCS 2# ;Br on EMT error
14 001626 000167 000356 JMP BSO1
15 001632 2#: FABORT EM$ILN ;Say invalid line number

```

Terminal

```

1          ;-----
2          ; SET TERMINAL AUTOBAUD
3          ;
4 001642  004767  0000000  SAUTO:  CALL  CKTERM      ;Require TERMINAL privilege
5 001646  052761  0000000  0000000  BIS    ##AUTO,LSW2(R1) ;Set flag saying to do autobaud select
6 001654  032761  0000000  0000000  BIT    ##DILUP,LSW(R1) ;Is anyone using the line now?
7 001662  001003                BNE    9#          ;Br if yes
8 001664  012705  0000000  MOV    #S9600,R5     ;Set line speed to 9600 baud
9 001670  000675                BR     STSP1
10 001672  000167  000312  9#:    JMP    BSO1

```

Terminal

```

1
2 ; -----
3 ; SET TERMINAL PARITY={EVEN ODD NONE}
4 SETPAR: MOV R4, -(SP) ; Save current option table pointer
5 CALL CKTERM ; Require TERMINAL privilege
6 CALL SKSPC ; Skip past any spaces
7 CMPB (R3), #'= ; Equal sign following PARITY?
8 BNE 1$ ; Br if not
9 INC R3 ; Skip past equal sign
10 1$: MOV #PARHD, R4 ; Point to new option table
11 CALL SETWRD ; Process EVEN, ODD, NONE word
12 MOV (SP)+, R4
13 RETURN

```

```

1
2 ;-----
3 ; SET TERMINAL BITS=m
4 001734 004767 0000000 STCLEN: CALL CKTERM ;Require TERMINAL privilege
5 001740 116105 0000010 MOVBL LMXPRM+1(R1),R5 ;Get current line parameters
6 001744 010102 MOV R1,R2 ;Save line index number
7 001746 004767 0000000 CALL ACRDEC ;Accrue parameter value
8 001752 042705 0000000 BIC #LP#7BT,R5 ;Assume 8 bits wanted
9 001756 020127 0000007 CMP R1,#7. ;Is value 7?
10 001762 001003 BNE 1$ ;Br if not
11 001764 052705 0000000 BIS #LP#7BT,R5 ;Select 7 bits
12 001770 000407 BR 2$
13 001772 120127 0000010 1$: CMPB R1,#8. ;Is value 8?
14 001776 001404 BEQ 2$ ;Br if yes
15 002000 FABORT #EM#ICL ;Invalid character length
16 002010 010201 2$: MOV R2,R1 ;Get back line index number
17 002012 000167 177446 JMP STSP1 ;Go set new length
    
```

Terminal

```

1
2 ; -----
3 ; SET TERMINAL PARITY={EVEN ODD NONE}
4 ;
5 ; Inputs:
6 ; R1 = line index number.
7 ; R4 = Pointer to work with parity flag bits.
8 ;
9 STPAR: CALL CKTERM ;Require TERMINAL privilege
10 MOVB LMXPRM+1(R1),R5 ;Get current flags for line
11 BIC #LP$PAR!LP$ODD,R5;Clear parity control flags
12 BISE @R4,R5 ;Set new flags
13 JMP STSP1 ;Go set value
14 ;
15 ; SET TERMINAL NOPARITY
16 ;
17 STNOPR: CALL CKTERM ;Require TERMINAL privilege
18 MOVB LMXPRM+1(R1),R5 ;Get current flags for line
19 BIC #LP$PAR!LP$ODD,R5;Clear parity control flags
20 JMP STSP1 ;Go set value
21 ; -----
22 ; SET TERMINAL [n] TRANSLATE={ext=int,ext=int,...}
23 ;
24 STTRNS: MOV R1,-(SP) ;Save original line number
25 MOV LNPRIM(R1),R1 ;Make sure we have primary line number
26 CALL PRSTRN ;Parse the translate qualifier
27 MOV (SP)+,R1 ;Restore original line number
28 RETURN

```

Terminal

```

1          ; -----
2          ; Set parameters for particular terminal types.
3          ;
4          ; SET PARAMETERS FOR A VT50/VT52.
5          ; SET VT50
6 002076 042761 0000000 0000000 SVT50: BIC      #VT52NO,LSW2(R1);CLEAR SOME FLAGS
7 002104 052761 0000000 0000000      BIS      #VT52FL,LSW2(R1);SAY TERMINAL IS A VT50
8 002112 012761 0000000 0000000      MOV      #VT52,LTRMTP(R1);SET TERMINAL TYPE
9 002120 000414          BR      BS02
10         ;
11        ; SET PARAMETERS FOR LA36.
12        ;
13 002122 042761 0000000 0000000 STLA36: BIC      #LA36NO,LSW2(R1);RESET VARIOUS FLAGS
14 002130 052761 0000000 0000000      BIS      #LA36FL,LSW2(R1)
15 002136 012761 0000000 0000000      MOV      #LA36,LTRMTP(R1);SET TERMINAL TYPE
16 002144 042761 0000000 0000000      BIC      ##SLON,LSW7(R1);Disable SL editor for this terminal type
17        ;
18        ; Set up terminal type information
19        ;
20 002152 120167 0000000          BSD2:  CMPB     R1,CORUSR      ;Changing terminal type for our job?
21 002156 001006          BNE      1$              ;Br if not our job
22 002160 005761 0000000          TST      LWINDO(R1)      ;Is process windowing turned on for job?
23 002164 001403          BEQ      1$              ;Br if not
24 002166 012700 000122'          MOV      #WINSTT,R0      ;Point to EMT argument block
25 002172 104375          EMT      375              ;Tell windowing system about terminal type
26 002174 105767 175622          1$:  TSTB     SETPRM      ;Should we make the set permanent?
27 002200 001403          BEQ      BS01              ;Br if not
28 002202 016161 0000000 0000000      MOV      LTRMTP(R1),ITRMTP(R1);Set initial terminal type
29 002210 016161 0000000 0000000 BSD01:  MOV      LSW2(R1),LSW2S(R1);Save in case of ctrl-c reentry
30 002216 105767 175600          TSTB     SETPRM      ;Should we make the set permanent?
31 002222 001403          BEQ      SETJMP      ;Br if not
32 002224 016161 0000000 0000000      MOV      LSW2(R1),ILSW2(R1);Set initial flags for line
33 002232 000207          SETJMP:  RETURN      ;Return to process next qualifier
34        ;
35        ; LA120 parameters
36        ;
37 002234 042761 0000000 0000000 SLA120: BIC      #LA12NO,LSW2(R1);SET FLAGS
38 002242 052761 0000000 0000000      BIS      #LA12FL,LSW2(R1)
39 002250 012761 0000000 0000000      MOV      #LA120,LTRMTP(R1);SET TERMINAL TYPE
40 002256 042761 0000000 0000000      BIC      ##SLON,LSW7(R1);Disable SL editor for this terminal type
41 002264 000732          BR      BS02
42        ;
43        ; VT100 parameters
44        ;
45 002266 042761 0000000 0000000 SVT100: BIC      #VT10NO,LSW2(R1);SET FLAGS
46 002274 052761 0000000 0000000      BIS      #VT10FL,LSW2(R1)
47 002302 012761 0000000 0000000      MOV      #VT100,LTRMTP(R1);SET TERMINAL TYPE
48 002310 000720          STERM:  BR      BS02
49        ;
50        ; VT200 parameters
51        ;
52 002312 042761 0000000 0000000 SVT200: BIC      #VT20NO,LSW2(R1)
53 002320 052761 0000000 0000000      BIS      #VT20FL,LSW2(R1)
54 002326 012761 0000000 0000000      MOV      #VT200,LTRMTP(R1)
55 002334 000706          BR      BS02
56        ;
57        ; Hazeltine parameters

```

```

58
59 002336 042761 0000000 0000000 STHAZL: BIC #HAZLND,LSW2(R1);RESET SOME FLAGS
60 002344 052761 0000000 0000000 BIS #HAZLFL,LSW2(R1);SET SOME FLAGS
61 002352 012761 0000000 0000000 MOV #HAZEL,LTRMTP(R1);SET TERMINAL TYPE
62 002360 042761 0000000 0000000 BIC ##SLON,LSW7(R1);Disable SL editor for this terminal type
63 002366 000671 BR BSG2
64
65 ; ADM3A parameters
66
67 002370 042761 0000000 0000000 SADM3A: BIC #ADM3ND,LSW2(R1);SET FLAGS
68 002376 052761 0000000 0000000 BIS #ADM3FL,LSW2(R1)
69 002404 012761 0000000 0000000 MOV #ADM3A,LTRMTP(R1);SET TERMINAL TYPE
70 002412 042761 0000000 0000000 BIC ##SLON,LSW7(R1);Disable SL editor for this terminal type
71 002420 000654 BR BSG2
72
73 ; SET PARAMETERS FOR DIABLO-1620 TERMINAL
74
75 002422 SQUME:
76 002422 042761 0000000 0000000 SDIAB: BIC #DIABND,LSW2(R1);RESET VARIOUS FLAGS
77 002430 052761 0000000 0000000 BIS #DIABFL,LSW2(R1);SAY THIS IS A DIABLO
78 002436 012761 0000000 0000000 MOV #DIABLO,LTRMTP(R1);SET TERMINAL TYPE
79 002444 042761 0000000 0000000 BIC ##SLON,LSW7(R1);Disable SL editor for this terminal type
80 002452 000637 BR BSG2
81
82 ; SET TT QUIET
83 ; (Also, SET NOVERIFY)
84
85 002454 116701 0000000 STNOVR: MOVB CORUSR,R1 ;Get current job index number
86 002460 004767 0000004 CALL SETQUT ;Do the set
87 002464 000167 0000000 JMP RDCMD ;Finished
88
89 002470 052761 0000000 0000000 SETQUT: BIS ##QTSET,LSW2(R1) ;REMEMBER SET WAS DONE
90 002476 052761 0000000 0000000 BIS ##QUIET,LSW4(R1) ;HAVE IMMEDIATE EFFECT
91 002504 120167 0000000 CMPB R1,CORUSR ;Are we doing set for current job?
92 002510 001025 BNE 1$ ;Br if not
93 002512 .GVAL #XAREA,#<CFSTS-MONVEC> ;GET CURRENT COMMAND FILE FLAGS
94 002532 010002 MOV RO,R2
95 002534 052702 0000000 BIS #CF$QUT,R2 ;SET TT-QUIET FLAG
96 002540 .PVAL #XAREA,#<CFSTS-MONVEC>,R2;STORE UPDATED FLAGS
97 002564 000167 177420 1$: JMP BSG1
98
99 ; SET TT NOQUIET
100 ; (Also, SET VERIFY)
101
102 002570 116701 0000000 STVRFY: MOVB CORUSR,R1 ;Get current job index number
103 002574 004767 0000004 CALL RSTQUT ;Do the SET
104 002600 000167 0000000 JMP RDCMD ;Finished
105
106 002604 042761 0000000 0000000 RSTQUT: BIC ##QTSET,LSW2(R1) ;REMEMBER SET WAS DONE
107 002612 042761 0000000 0000000 BIC ##QUIET,LSW4(R1) ;START LISTING NOW
108 002620 120167 0000000 CMPB R1,CORUSR ;Are we doing set for current line?
109 002624 001025 BNE 1$ ;Br if not
110 002626 .GVAL #XAREA,#<CFSTS-MONVEC> ;GET CURRENT COMMAND FILE FLAGS
111 002646 010002 MOV RO,R2
112 002650 042702 0000000 BIS #CF$QUT,R2 ;CLEAR TT-QUIET FLAG
113 002654 .PVAL #XAREA,#<CFSTS-MONVEC>,R2;STORE UPDATED FLAGS
114 002700 000167 177304 1$: JMP BSG1

```

Terminal

```

115 ;
116 ; SET TT NOSCOPE
117 ;
118 002704 042761 0000000 0000000 SETNSC: BIC    ##SCOPE,LSW2(R1) ;Say this is not a scope type terminal
119 002712 042761 0000000 0000000          BIC    ##SLON,LSW7(R1) ;Disable SL editor
120 002720 000167 177264          JMP     BSO1
121 ;
122 ; SET TT WAIT
123 ;
124 002724 042761 0000000 0000000 SETTW:  BIC    ##SNWTT,LSW6(R1);UNDO SET TT NOWAIT
125 002732 042761 0000000 0000000          BIC    ##NOWTT,LSW5(R1)
126 002740 000207          RETURN
127 ;
128 ; SET TT NOWAIT
129 ;
130 002742 052761 0000000 0000000 SETTNW: BIS    ##SNWTT,LSW6(R1);SET TT NOWAIT
131 002750 052761 0000000 0000000          BIS    ##NOWTT,LSW5(R1)
132 002756 000207          RETURN
133 ;
134 ; SET TT DEAD
135 ;
136 002760 004767 0000000          SETDED: CALL   CKTERM          ;Require TERMINAL privilege
137 002764 032761 0000000 0000000          BIT    ##DILUP,LSW(R1) ;Is line active now
138 002772 001406          BEQ     SETTTB          ;Br if not
139 002774          FABORT  #RUNMS          ;Can't kill an active line
140 ;
141 ; Perform terminal sets that only involve setting or clearing flag bits.
142 ;
143 003004 004767 0000000          SETTTP: CALL   CKTERM          ;Require TERMINAL privilege
144 003010 012405          SETTTB: MOV    (R4)+,R5        ;POINT TO TABLE
145 003012 060105          ADD     R1,R5              ;POINT TO TABLE ENTRY FOR THIS USER
146 003014 051415          BIS    @R4,@R5          ;SET THE DESIRED FLAG
147 003016 000167 177166          JMP     BSO1
148 ;
149 ; RESET A BIT IN A USER TABLE
150 ;
151 003022 004767 0000000          CLR'TTP: CALL  CKTERM          ;Require TERMINAL privilege
152 003026 012405          CLR'TTB: MOV   (R4)+,R5        ;POINT TO TABLE
153 003030 060105          ADD     R1,R5              ;POINT TO TABLE ENTRY FOR THIS USER
154 003032 041415          BIC    @R4,@R5          ;RESET THE DESIRED FLAG
155 003034 000167 177150          JMP     BSO1

```

CL

```

1
2
3
4
5
6
7
8 003040 004767 0000000 SETCL: CALL CKTERM ;Require TERMINAL privilege for SET CL
9 003044 020227 0000000 CMP R2,#CLTOIL ;Is this a valid unit?
10 003050 103404 BLO 3$ ;Br if ok
11 003052 FABORT #EM$IUN ;Invalid CL unit
12 003062 006302 3$: ASL R2 ;Convert unit # to word table index
13
14 ; At this point, R2 contains the index into the CL unit tables.
15 ; Begin loop to process each specified command qualifier.
16
17 003064 004767 0000000 1$: CALL SKPSPC ;Skip over any spaces
18 003070 111300 MOV (R3),R0 ;Get next character from command
19 003072 001434 BEQ 15$ ;Br if hit the end of the command
20 003074 120027 000054 CNPB R0,#',' ;Comma separator?
21 003100 001403 BEQ 4$ ;Br if yes
22 003102 120027 000057 CNPB R0,#'/' ;Slash separator?
23 003106 001000 BNE 5$ ;Br if not
24 003110 005203 4$: INC R3 ;Skip past separator
25 003112 004767 0000000 CALL SKPSPC ;Skip over any spaces
26 003116 105713 TSTB (R3) ;Are we at the end of the command now?
27 003120 001410 BEQ 10$ ;Br if yes -- Error
28
29 ; Process the next command qualifier
30
31 003122 012704 001120' 5$: MOV #CLOPHD,R4 ;Point to option table
32 003126 004767 0000000 CALL SEARCH ;Search for option keyword
33 003132 103404 BCS 11$ ;Br if cannot identify keyword
34 003134 010246 MOV R2,-(SP) ;Save unit index
35 003136 004734 CALL @(R4)+ ;Call routine to process qualifier
36 003140 012602 MOV (SP)+,R2 ;Get back unit number index
37 003142 000750 BR 1$ ;Go see if there are more qualifiers
38
39 ; Invalid qualifier keyword
40
41 003144 11$: FABORT #CSIMS1 ;Invalid option keyword
42
43 ; Invalid command syntax
44
45 003154 10$: FABORT #EM#CSE ;Command syntax error
46
47 ; End of command reached
48
49 003164 000167 0000000 15$: JMP RDCMD ;Finished command
50
51 ; CL unit is not assigned to a line
52
53 003170 STCLNL: FABORT #EM#CLN ;CL unit is not assigned to a line
54
55 ; Turn on some option
56
57 003200 005762 0000000 STCLOP: TST CL$LIX(R2) ;Is this CL unit assigned to a line?

```

CL

```

58 003204 001771          BEQ     STCLNL          ;Br if not
59 003206 051462 0000000  BIS     (R4),CL#OPT(R2) ;Set desired option flag
60 003212 000207          RETURN
61
62          ; Turn off some option
63
64 003214 005762 0000000  CLCLOP: TST     CL$LIX(R2) ;Is this CL unit assigned to a line?
65 003220 001763          BEQ     STCLNL          ;Br if not
66 003222 041462 0000000  BIC     (R4),CL#OPT(R2) ;Reset the option
67 003226 000207          RETURN
68
69          ; Set a parameter value
70
71 003230 005762 0000000  STCLVL: TST     CL$LIX(R2) ;Is this CL unit assigned to a line?
72 003234 001755          BEQ     STCLNL          ;Br if not
73 003236 004767 0000000  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 0000000  STCLR:  TST     CL$LIX(R2) ;Is this CL unit assigned to a line?
82 003256 001744          BEQ     STCLNL          ;Br if not
83 003260 011462 0000000  MOV     (R4),CL#WID(R2) ;Set width to 0 (graph) or 132 (nograph)
84 003264 000207          RETURN
85
86          ; SET CL TOP
87
88 003266 005762 0000000  CLTOP:  TST     CL$LIX(R2) ;Is this CL unit assigned to a line?
89 003272 001736          BEQ     STCLNL          ;Br if not
90 003274 005062 0000000  CLR     CL$LIN(R2)      ;Say we are at top of the page
91 003300 000207          RETURN
92
93          ; SET CL SPEED=value
94
95 003302 016202 0000000  STCLSP: MOV     CL$LIX(R2),R2 ;Is this CL unit assigned to a line?
96 003306 001730          BEQ     STCLNL          ;Br if not
97 003310 004767 0000000  CALL    ACRSPD          ;Accrue speed value
98 003314 116200 0000010  MOVB   LMXPRM+1(R2),R0 ;Get current flags for line
99 003320 042700 0000000  BIC     #LP$SPD,R0      ;Clear speed code
100 003324 050005          BIS     R0,R5          ;Put in parity and length flags
101 003326 010567 174530  STCLSI: MOV     R5,LSPEMT+4 ;Store speed code into EMT arg block
102 003332 006202          ASR     R2              ;Convert line index # to line #
103 003334 010267 174520  MOV     R2,LSPEMT+2     ;Store line # into EMT arg block
104 003340 012700 000056' MOV     #LSPEMT,R0      ;Point to EMT arg block
105 003344 104370          EMT     375            ;Set the speed
106 003346 103401          BCS    2$             ;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

```

CL

```

115 003362 012705 0000000 STCLXN: MOV #CLSFCH,R5 ;Get .SPFUN code
116 003366 000402 BR CLSPFN ;Do the .SPFUN
117 ;
118 ; SET CLn RESET
119 ;
120 003370 012705 0000000 STCLRS: MOV #CLSFRS,R5 ;Get .SPFUN code
121 ;
122 ; Do a .SPFUN to a CL unit.
123 ; R5 = .SPFUN code.
124 ;
125 003374 005762 0000000 CLSPFN: TST CL$LIX(R2) ;Is this CL unit assigned to a line?
126 003400 001673 BEQ STCLNL ;Br if not
127 003402 004767 001040 CALL CLLOOK ;Open channel 1 to CL unit
128 003406 .SPFUN #XAREA,#1,R5,#0,#0 ;Do the function
129 003452 103404 BCS 2# ;Br if error on .SPFUN
130 003454 .CLOSE #1 ;Close CL channel
131 003462 000207 RETURN
132 003464 2#: .PURGE #1 ;Make sure channel is closed
133 003472 FABORT #EM#IUN ;Say invalid unit number
134 ;
135 ; SET CL PARITY driver routine
136 ;
137 003502 004767 0000000 STCLPD: CALL SKPSPC ;Skip any spaces
138 003506 122327 000075 CMPB (R3)+,#/= ;Should have an equal sign
139 003512 001401 BEQ 1#
140 003514 005303 DEC R3 ;Point to start of keyword
141 003516 012704 001556' 1#: MOV #CLPRHD,R4 ;Point to driver table
142 003522 004767 0000000 CALL SETWRD ;Process the option
143 003526 000207 RETURN

```

CL

```

1
2 ; -----
3 ; SET CL PARITY=(EVEN ODD NONE)
4 ;
5 ; Inputs:
6 ; R2 = CL unit number index
7 ; R4 = Pointer to word with parity control flags
8 003530 011404 STCLPR: MOV (R4),R4 ;Get parity flags
9 003532 016202 0000000 STCLP1: MOV CL$LIX(R2),R2 ;Get number of line CL unit is connected to
10 003536 001406 BEQ 9$ ;Br if not assigned to a line
11 003540 116205 0000010 MOVE LMXPRM+1(R2),R5 ;Get current control flags for line
12 003544 042705 0000000 BIC #LP#PAR!LP#ODD,R5 ;Clear parity control flags
13 003550 050405 BIS R4,R5 ;Put in new flags
14 003552 000660 BR STCLS1 ;Go set value for line
15 003554 9$: FABORT #EM#CLN ;CL unit is not assigned to a line
16 ;
17 ; SET CL NOPARITY
18 ;
19 003564 005004 STCLNP: CLR R4 ;Clear parity flags
20 003566 000761 BR STCLP1 ;Go do the set

```

CL

```

1
2 ; -----
3 ; SET CL BITS=11
4 003570 016202 0000000 STCLBT: MOV CL#LIX(R2),R2 ;Get index for line CL unit is assigned to
5 003574 001420 BEQ 9# ;Br if not assigned to a line
6 003576 004767 0000000 CALL ACRDEC ;Accrue parameter value
7 003602 116205 0000010 MOVB LMXPRM+1(R2),R5 ;Get current flags for line
8 003606 042705 0000000 BIC #LP#7BT,R5 ;Assume 8 bits wanted
9 003612 020127 0000007 CMP R1,#7. ;7 bits wanted?
10 003616 001003 BNE 1# ;Br if not
11 003620 052705 0000000 DIS #LP#7BT,R5 ;Set 7 bit flag
12 003624 000407 BR 2#
13 003626 020127 000010 1#: CMP R1,#8. ;8 bits wanted?
14 003632 001404 BEQ 2# ;Br if yes
15 003634 FABORT #EM#ICL ;Invalid character length
16 003644 000167 177456 2#: JMP STCLS1 ;Go set value
17 003650 9#: FABORT #EM#CLN ;CL unit not assigned to line

```

CL

```

1 ; -----
2 ; SET CL LINE=TL
3 ;
4 003660 STCLLN:
5 ;
6 ; Accrue the line number
7 ;
8 003660 004767 0000000 CALL ACRDEC ;Accrue decimal value
9 ;
10 ; Do the EMT to assign this CL unit to a line and check for errors.
11 ;
12 003664 006202 ASR R2 ;Convert CL unit index to unit #
13 003666 010267 174152 MOV R2,CLAEMT+2 ;Set CL unit number
14 003672 010167 174150 MOV R1,CLAEMT+4 ;Set time-sharing line number
15 003676 012700 000042' MOV #CLAEMT,R0 ;Point to EMT argument block
16 003702 104370 ENT 375 ;Do the EMT
17 003704 103045 BCC 20$ ;Br if no error
18 ;
19 ; An error occurred on the EMT.
20 ; Print an error message
21 ;
22 003706 113702 0000000 MOVB @#ERRLOC,R2 ;Get EMT error code
23 003712 120227 000002 CNPB R2,#2. ;Invalid CL unit # ?
24 003716 001004 BNE 3$ ;Br if not
25 003720 FABORT #EM$IUN ;Invalid CL unit number
26 003730 120227 000003 3$: CNPB R2,#3. ;Invalid line number?
27 003734 001004 BNE 4$ ;Br if not
28 003736 FABORT #EM$ILN ;Invalid line number
29 003746 120227 000004 4$: CNPB R2,#4. ;Line already in use by a CL unit?
30 003752 001004 BNE 5$ ;Br if not
31 003754 FABORT #EM$ACL ;Line already being used by CL
32 003764 120227 000005 5$: CNPB R2,#5. ;Is line in use by a time-sharing user?
33 003770 001004 BNE 6$ ;Br if not
34 003772 FABORT #EM$TSL ;Line busy as time-sharing line
35 004002 120227 000006 6$: CNPB R2,#6. ;Is this CL unit currently busy?
36 004006 001004 BNE 20$ ;Br if not
37 004010 FABORT #EM$CLB ;This CL unit is busy
38 ;
39 ; Finished
40 ;
41 004020 000207 20$: RETURN

```

CL

```

1
2 ; -----
3 ; GET CL VERSION=n
4 004022 004767 0000000 STCLVR: CALL ACRDEC ; Accrue parameter value
5 004026 120127 000017 CMPB R1,#15 ; Is this a valid argument
6 004032 002403 BLT 9$ ; Br if not
7 004034 110167 0000000 MOVE R1,CLVERS ; Set version number
8 004040 000404 BR 10$ ; Return
9 004042 9$: FABORT #EM$IVN ; Invalid version number
10 ;
11 ; Finished
12 ;
13 004052 000207 10$: RETURN
14

```

CL

```

1
2 ; -----
3 ; SET CLn ENDSTRING='string'
4 004054 005762 0000000 CLESTR: TST CL$LIX(R2) ;Is this CL unit assigned to a line?
5 004060 001000 BNE 2$ ;Br if yes
6 004062 000167 177102 JMP STCLNL ;Br if not
7 004066 010446 2$: MOV R4, -(SP)
8 004070 010546 MOV R5, -(SP)
9 ;
10 ; Accrue the string
11 ;
12 004072 004767 0000000 CALL ACRSTR ;Accrue the string
13 ;
14 ; Make sure the string is not too long
15 ;
16 004076 020027 0000000 CMP RO, #CLEDFS ;Is string too long?
17 004102 101046 BHI 10$ ;Br if yes
18 ;
19 ; Move string over in BLKO to allow room for ENDPAGE value in 1st word
20 ;
21 004104 012705 0000000 MOV #BLKO, R5 ;Point to start of buffer
22 004110 005200 INC RO ;Get length of string with null
23 004112 060000 ADD RO, R5 ;Point past last char in string
24 004114 010504 MOV R5, R4
25 004116 062704 0000002 ADD #2, R4 ;Add 1 word offset
26 004122 114544 3$: MOVB -(R5), -(R4) ;Move string over
27 004124 077000 SOB RO, 3$
28 004126 012767 000377 0000000 MOV #377, BLKO ;Set value for ENDPAGE (377=no change)
29 ;
30 ; Do lookup on CL unit
31 ;
32 004134 004767 000306 CALL CLLOOK ;Do lookup on channel 1 to CL unit
33 ;
34 ; Perform the .SPFUN to set the ENDSTRING
35 ;
36 004140 .SPFUN #XAREA, #1, #CLSFEF, #BLKO, #10., #0
37 004204 .CLOSE #1 ;Close the channel
38 ;
39 ; Finished
40 ;
41 004212 012605 MOV (SP)+, R5
42 004214 012604 MOV (SP)+, R4
43 004216 000207 RETURN
44 ;
45 ; String is too long
46 ;
47 004220 10$: FABORT #EM$STL ;String too long

```

CL

```

1
2 ; -----
3 ; SET CLn TRANSLATE=(ext=int,ext=int,...)
4 004230 016201 0000000 CLTRNS: MOV CL#LIX(R2),R1 ;Is this CL unit assigned to a line?
5 004234 001002 BNE 2# ;Br if yes
6 004236 000167 176726 JMP STCLNL ;Br if not
7 004242 004767 000002 2#: CALL PRSTRN ;Parse the translate qualifier
8 004246 000207 RETURN
9
10 ; -----
11 ; Parse a translate qualifier of the form:
12 ; TRANSLATE=(ext=int,ext=int,...)
13 ; and store the translation table for the line.
14 ;
15 ; Inputs:
16 ; R1 = Line index number of line whose translation table is being set up.
17 ; R3 = Pointer past end of keyword
18 ;
19 004250 010446 PRSTRN: MOV R4,-(SP)
20 004252 010546 MOV R5,-(SP)
21 ;
22 ; Get pointer to translation table area for this line
23 ;
24 004254 016104 000000G MOV LCXTBL(R1),R4 ;Get pointer to translation table for line
25 004260 001004 BNE 7# ;Br if there is one
26 004262 FABORT #EM$TMT ;No translation table allocated
27 ;
28 ; See if equal sign and open paren follow keyword
29 ;
30 004272 004767 000000G 7#: CALL SKPSPC ;Skip over any spaces
31 004276 122327 000075 CMPB (R3)+,#'=' ;Does equal sign follow keyword?
32 004302 001406 BEQ 1# ;Br if yes
33 004304 126327 177777 000050 CMPB -1(R3),#'(' ;Did user omit equal sign?
34 004312 001413 BEQ 2# ;Br if yes
35 004314 005303 DEC R3 ;Point back to character
36 004316 000447 BR 3# ;Clear the translation table
37 004320 004767 000000G 1#: CALL SKPSPC ;Skip more spaces
38 004324 122327 000050 CMPB (R3)+,#'(' ;Should have open paren
39 004330 001404 BEQ 2# ;Br if got open paren
40 004332 FABORT #EM$CSE ;Invalid syntax
41 ;
42 ; Begin to accrue each value pair
43 ;
44 004342 005005 2#: CLR R5 ;Count pairs in R5
45 ;
46 ; See if we have reached the end of the list
47 ;
48 004344 004767 000000G 5#: CALL SKPSPC ;Skip any spaces
49 004350 122327 000051 CMPB (R3)+,#')' ;At end of the list?
50 004354 001430 BEQ 3# ;Br if yes
51 004356 005303 DEC R3 ;Point back to character
52 ;
53 ; See if we have room for another value
54 ;
55 004360 020527 000000G CMP R5,#MXTTCT ;Room for another value?
56 004364 103404 BLD 4# ;Br if yes
57 004366 FABORT #EM$TMT ;Too many translation values

```

CL

```

58 004376 005205      4$:      INC      R5      ;Count another value
59                    ;
60                    ; Accrue the external value
61                    ;
62 004400 004767 0000000      CALL      ACROCT      ;Accrue the external value
63 004404 110164 000001      MOV      R1,1(R4)    ;Store in high-order byte of word
64                    ;
65                    ; Accrue the internal value
66                    ;
67 004410 004767 0000000      CALL      ACROCT      ;Accrue the internal value
68 004414 110114      MOV      R1,(R4)    ;Store in low-order byte of word
69 004416 005724      TST      (R4)+      ;Point to next word
70 004420 004767 0000000      CALL      SKPSPC     ;Skip any spaces
71 004424 122327 000054      CNPB     (R3)+,#',  ;Comma separator?
72 004430 001745      BEQ      5$         ;Br if yes
73 004432 005303      DEC      R3         ;Point back to char we skipped
74 004434 000743      BR      5$         ;Loop and get rest of list
75                    ;
76                    ; We have reached the end of the list
77                    ;
78 004436 005014      3$:      CLR      (R4)      ;Store a zero to terminate the list
79                    ;
80                    ; Finished
81                    ;
82 004440 012605      MOV      (SP)+,R5
83 004442 012604      MOV      (SP)+,R4
84 004444 000207      RETURN

```

CL

```

1      ;-----
2      ; Perform a lookup on channel 1 to a specified CL unit.
3      ;
4      ; Inputs:
5      ;   R2 = CL unit index number.
6      ;
7      ; Outputs:
8      ;   Channel 1 opened to CL unit.
9      ;
10     004446  010246  CLLOOK: MOV      R2, -(SP)
11     ;
12     ; Construct CL device name
13     ;
14     004450  006207          ASR      R2          ;Get cl unit number
15     004452  020227  000007  CMP      R2, #7.      ;Should this be a CL or C1 unit?
16     004456  101405          BLOS    2#          ;Br if CL
17     004460  162702  000010  SUB     #8, R?       ;Remove C1 unit bias
18     004464  066702  173326  ADD     R5OC10, R2   ;Add "C10" to form device name
19     004470  000402          BR     1#          ;
20     004472  066702  173316  2#:   ADD     R5OCLO, R? ;Add "CLO" to form device name
21     004476  010267  173302  1#:   MOV     R2, CLDEV ;Save device name
22     004502          .LOOKUP #XAREA, #1, #CLDEV; Open channel to CL unit
23     004522  103402          BCS   10#         ;Br if error on lookup
24     ;
25     ; Finished
26     ;
27     004524  012607          MOV     (SP)+, R2
28     004526  000207          RETURN
29     ;
30     ; Error on lookup to CL unit
31     ;
32     004530  10#:   FABORT #EM$IUN ;Invalid unit number

```

```

Host
1
2
3
4
5
6 004540 004767 0000000
7
8
9
10 004544 012767 177777 173302
11 004552 012704 001604
12 004556 004767 0000000
13
14
15
16
17 004562 016705 173266
18 004566 020527 0000000
19 004572 103404
20 004574
21 004604 006305
22 004606 010567 173242
23 004612 005765 0000000
24 004616 001004
25 004620
26 004630 010500
27 004632 004767 0000000
28 004636 006300
29 004640 001407
30 004642 120067 0000000
31 004646 001404
32 004650
33
34
35
36 004660 116701 0000000
37 004664 016761 173110 0000000
38 004672 016761 173104 0000000
39 004700 012700 000050
40 004704 104375
41
42
43
44 004706 016761 0000000 0000000
45 004714 016761 0000020 0000000
46 004722
47 004730 000167 0000000
48
49
50
51 004734 010246
52 004736 004767 0000000
53 004742 004767 0000000
54 004746 122327 000072
55 004752 001401
56 004754 005303
57 004756 016700 0000000

          . SBTTL          Host
-----
; SET HOST/DTE=n
; Cross connect time-sharing line with CL unit.
;
SETHST: CALL    CKTERM          ;Require TERMINAL privilege for SET HOST
;
; Do command scanning
;
          MOV      #-1,TTXCL+4    ;Initially set CL unit to -1
          MOV      #HOSTHD,R4     ;Point to option driver table
          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.
;
          MOV      TTXCL+4,R5     ;Get CL unit number
          CMP      R5,#CLTOTL     ;Is this a valid unit number
          BLO     1$             ;Br if yes
          FABORT   #EM#IUN        ;Invalid CL unit number
          ASL     1$             ;Convert to CL index number
          MOV      R5,TTXCL+4
          TST     CL#LIX(R5)      ;Is this CL unit connected to a line?
          BNE     2$             ;Br if yes
          FABORT   #EM#CLN        ;Not connected to line
          MOV      R5,R0          ;Get CL unit index
          CALL    CKCLUS         ;See if CL unit in use by another job
          ASL     R0              ;Any job using CL unit?
          BEQ     3$             ;Br if not
          CMPB    R0,CORUSR      ;Is it our job?
          BEQ     3$             ;Br if yes
          FABORT   #EM#CLB        ;This CL unit is busy
;
; Perform the connection
;
          MOV      CORUSR,R1      ;Get our job index number
          MOV      R5OHST,LPR01(R1);Set name of running program to "$HOST$"
          MOV      R5OHST+2,LPR02(R1)
          MOV      #TTXCL,R0      ;Point to EMT argument block
          EMT     375             ;Make the cross connection
;
; Print message saying cross connection broken
;
          MOV      R5OKMN,LPR01(R1);Reset program name to KMON
          MOV      R5OKMN+2,LPR02(R1)
          .PRINT   #TM#XIK
          JMP      RDCMD          ;Finished with command
;
; Process the /PORT=udn qualifier
;
HSTPRT: MOV      R2,-(R3)
          CALL    CHKEQ          ;Equal sign should follow qualifier name
          CALL    QTRD50         ;Accrue the device name
          CMPB    (R3)+,#/:      ;Colon specified with device name?
          BEQ     1$             ;Br if yes
          DEC     R3              ;Backup pointer
          MOV     R50BUF,R0      ;Get accrued device name
          1$:

```

Host

```

58 004762 004767 0000000 CALL ASNSRC ;See if this is a logical device name
59 004766 103402 BCS 2$ ;Br if not logical name
60 004770 016200 0000000 MOV AT$DEV(R2),R0 ;Get physical device name
61 004774 004767 0000000 2$: CALL CHKCLU ;Convert CL and C1 names to unit numbers
62 005000 103404 BCS 10$ ;Br if not CL or C1 unit
63 005002 010067 173046 MOV R0,TTXCL+4 ;Set CL unit number
64 005006 012602 MOV (SP)+,R2
65 005010 000207 RETURN
66 005012 10$: FABORT #EM$CLX ;Port unit must be CL or C1 unit
67 ;
68 ; Process the /DTE=n SET HOST qualifier
69 ;
70 005022 010146 HSTDTE: MOV R1,-(SP)
71 ;
72 ; Accrue the CL unit number
73 ;
74 005024 004767 0000000 CALL ACRDEC ;Accrue the CL unit number
75 005030 010167 173020 MOV R1,TTXCL+4 ;Store CL unit index in EMT arg block
76 005034 012601 MOV (SP)+,R1
77 005036 000207 RETURN
78 ;
79 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						
\$BRIT	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						
\$CFCCCL	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	5-7	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						
\$JNDAB	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						
\$NDIN	1-72						
\$NDINT	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				
\$RNIDP	1-205						
\$SCOPE	1-111	5-44	23-118				
\$SCALL	1-124						
\$SCHIO	1-123						

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											
INV DAT	1-200											
INVDEV	1-199											
INVEC	1-158											
INV LDM	1-195											
INV LDN	1-172											
INVOPT	1-165	1-171		1-186								
INVTIM	1-191											
IOABFL	1-61											
ITR MTP	1-159	23-23*										
JCXP GS	1-142											
JCX SMS	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											
KMNP GS	1-79											
KMNSTK	1-79											
KMNSTR	1-79											
KMNTOP	1-79											
KMPRMT	1-138											
L	5-5	5-10#	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#

Cross reference table (CREF V05.04)

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							
LDCKTX	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							
LDMAP	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							
LSECT	1-95							
LSPENT	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						

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
PHMEM	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		
P RTPCT	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#	11-30*	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		
QUMEND	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		

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	
TKIVAL	1-141	
TM#AUT	1-57	
TM#CDS	1-56	
TM#CEN	1-56	
TM#CLO	1-58	
TM#CL1	1-58	
TM#CL2	1-58	
TM#CL3	1-58	
TM#CL4	1-58	
TM#CL5	1-58	
TM#CL6	1-58	
TM#CNG	1-56	
TM#GBL	1-49	
TM#HPE	1-56	
TM#HPR	1-55	
TM#IN1	1-38	
TM#IN2	1-38	
TM#LCL	1-49	
TM#LPR	1-55	
TM#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	
TMIDWH	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#	1-73
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						
VCDRTM	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						
VQUANO	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					
VT10ND	1-149	23-45					
VT200	1-146	23-54					
VT2007	1-146						
VT2008	1-146						
VT20FL	1-150	23-53					
VT20ND	1-150	23-52					
VT52	1-147	23-8					
VT52FL	1-148	23-7					
VT52ND	1-126	23-6					

Cross reference table (CREF V05.04)

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
XDNEMT	4-42#	17-5*	17-6*	17-7				
YESTXT	1-178							
ZCLR	1-156							

... 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	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#												
. TOUT	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#	8-27	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						