Table of contents

```
                                        .TITLE  TSKm2A -- Keyboard command routines
                                        .ENABL  LC
                                        .DSABL  GBL
000000                                  .CSECT  TSKM2A
000000                          TSKM2A:
                                ;
                                ;  TSKM2A is the portion of TSKMON that contains the actual code
                                ;  to implement each keyboard command that is processed by TSKMON.
                                ;
                                ;  Copyright 1978, 1979, 1980, 1981, 1982, 1983, 1984.
                                ;  S&H Computer Systems, Inc.
                                ;  Nashville, Tennessee
                                ;
                                ;  Macro calls
                                ;
                                        .MCALL  .CSISPC,.TTOUTR,.SRESET,.CRRG,.ELRG
                                        .MCALL  .READW,.TTYIN,.TTYOUT,.PURGE,.TWAIT
                                        .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  TSKM2A,CMDHD,CMDOFF,KDOCIN
                                        .GLOBL  PLOAD,CMDFRM,CMDDSN,STLGCN,DATTIM,PRGALL
                                        .GLOBL  DLCEMT,ALCDEV,CMDINS,CMDSPN,CMDRSM,CMDYEL
                                        .GLOBL  CMDRSY,CMDRST,CMDSND,CMDFMT,CMDDAT
                                        .GLOBL  CMDTIM,CMDMNT,CMDMON,CMDDMT,CMDDSP,CMDASN
                                        .GLOBL  CMDALC,CMDDLC,OPRCMD,CMDSPO,CMDKIL,CMDREM
                                        .GLOBL  CMDPAU,CMDDET,CMDACC,CMDINI,CMDSQZ
                                        .GLOBL  CMDBOT,CMDSHT
                                ;
                                ;  Global references
                                ;
                                        .GLOBL, TM$SA1,TM$SA2,TM$SA3,TM$SA4,SKPSPC,SKPDLM
                                        .GLOBL  SWPCHN,SEGCHN,INDTSV,LWINDO,$NOWIN,LSW11
                                        .GLOBL  $SCCA,AF$CCA,AFCF,EM$NUC,PO$SYS,AF$NPW
                                        .GLOBL  AF$DUP,AF$IND,AF$UCL,AF$SET,INSSRC,FORCEO
                                        .GLOBL  CFSTOP,CFSTRT,EM$ALC,EM$NFW,EM$SND,EM$DNR
                                        .GLOBL  CFSQEZ,TOOLNG,SDFHD,AF$TPO,$TRNSP,EM$CSE,WLDNAM
                                        .GLOBL  EM$NAL,AF$PLK,AF$DBG,FIXPRV,EM$LDI,EM$NLN
                                        .GLOBL  PO$MEM,PO$NFW,PO$BYP,PO$ALC,CKSYPV,CKACOJ,PRIVSO
                                        .GLOBL  PFSO,PFCO,PVNPW,EM$NAD,PO$DBG,PRIVCO,II$PRV,II$NPV
                                        .GLOBL  CLRPRV,EM$FNI,EM$ITF,INPADR,INPEMT,OPTLST
                                        .GLOBL  CDBUF,CDGET,INGADR,INGEMT,IIBUF,PRVOPT
                                        .GLOBL  CINFLG,$VNOTT,SDNAME,DMYDEV
                                        .GLOBL  CORUSR,LSW,$CTRLO,SERFLG,IOABFL
                                        .GLOBL  LSTHL,LCLUNT,FSTIOL,LSTIOL,HANCHN
                                        .GLOBL  MAXALC
                                        .GLOBL  AD$DVU,AD$JOB,AD$$SZ,LSTSPL,SOPALC
                                        .GLOBL  NEDCHR,LOUTIR,LINIR,LINRTS,CLOTIR
                                        .GLOBL  LCDTYP,SOPDAT,SOPTIM,PO$LOK
                                        .GLOBL  UTRPAD,JSWLOC,ERRLOC,MAXMEM,MAXPRI
                                        .GLOBL  PO$MEM,PO$DET,EM$DET,PO$SND,EM$MPV,P2$WRL
                                        .GLOBL  USRSTK,$KINIT,CFSTK,MXJMEM,DFJMEM,LPARNT
```

```
                .GLOBL  SPUBUF,SXBPNT,MXJADR
                .GLOBL  TMTOTH,TMTOTL,TMUSRH,TMIOWH,LDMNT
                .GLOBL  R.GSIZ,RS.GBL,RS.PVT,R.NAME,RS.CRR,RS.EGR
                .GLOBL  EM$UAR,EM$UER,R.GSTS,RSTPRV
                .GLOBL  TMSWTH,TMIDLH,TMIOH,TMSWPH,LDCLEN
                .GLOBL  EM$NPD,EM$IAD,EM$ATF,EM$IAD,EM$NLD
                .GLOBL  EM$SSY,EM$NID,QHDMS1,QHDMS2,EM$NSF,EM$OLO
                .GLOBL  GENMON,JS$OFF,LMONHD,AF$MEM,AF$BYA,SMONHD
                .GLOBL  EM$DAA,EM$DIU,CHKALC,$CHACT,$STSNG
                .GLOBL  WILDFL,$NOIN,$NOWTT,$HITTY
                .GLOBL  TECO,EDIT,KED,K52,$1STLG,$DIBOL,AF$SCA
                .GLOBL  LINBUF,LINNXT,LSTACT,PRGTOP,PRGSIZ
                .GLOBL  CXTPAG,FSTIOL
                .GLOBL  LAFSIZ,LFWLIM,LINCUR,NUMON,ILSW2
                .GLOBL  $DBKMN
                .GLOBL  $CARUP,DOASGN,UKMNAM,$UKMON,LSW9
                .GLOBL  LSUCF,$CCLRN,EM$NUK
                .GLOBL  KL3CLR,$PRGLK,LSW5,PVON
                .GLOBL  ALDEMT,TALEMT
                .GLOBL  LSTDL,FSTDL,$DETCH,UMSYTP
                .GLOBL  $DISCN,LPROJ,LPROG,LUNAME
                .GLOBL  LCPUHI,LCPULO,LCONTM,$CTRLS,$SPLJB
                .GLOBL  STPFLG,TOTON,USPLCH,SPLCHN
                .GLOBL  S$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,CLSFSP
                .GLOBL  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,$CFABT,INDSTA,INDERR
                .GLOBL  LNBLKS,CXTBAS,CXTWDS,UHIMEM
                .GLOBL  ASNTBL,$DILUP,CSHDEV,CSHDVN,LNSBLK
                .GLOBL  AT$LOG,AT$SIZ,AT$DEV,AT$FIL,AT$$SZ,AT$EXT
                .GLOBL  ASNEND,LSW3,LSW2S,$DUPRN
                .GLOBL  $FORM,$TAB,LSCCA,$CFSOT,LOFSPC,R50COM
                .GLOBL  $PAGE,$SCOPE,$ECHO,$LC,$SETRN,EM$FOE
                .GLOBL  UCHAN,$FORMO,$CFALL,$CFDCC,$CFCCL
                .GLOBL  LNPRIM,LNMAP,CW$SOH,CONFIG,$SUCF
                .GLOBL  $DOOFF,NUCHN,LRBFIL,CFIND
                .GLOBL  C.CSW,C.DEVQ,C.SBLK,NLINES
                .GLOBL  CD$NAM,CD$DVU,CD$BAS,CD$JOB,CD$$SZ,CD$$UB
                .GLOBL  LTSCMD,LNSPAC,CFNEST,VSWPFL,UCLNAM
                .GLOBL  $CFOPN,CFSEND,PBFEND,CFSP,$TTGAG
                .GLOBL  UFPTRP,SDSFCB,SD$DEL,CFLFL4,$UCLCF
                .GLOBL  SDFLAG,SD$FLK,SD$WFM,SDFORM,$UCLRN
                .GLOBL  SDBUF1,SDBLK,SPLND,LD$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,$QUIET
                .GLOBL  LSW4,KL4CLR,SDSKIP,SDBU,SD$BAK
                .GLOBL  $INCOR,$KED,VQUN1B,VINTIO,VQUN1C
                .GLOBL  SFFORM,SD$SNG,SFNMBL,NFRESB
```

```
                        .GLOBL  SD$HLD, CURPRM, PRMPNT
                        .GLOBL  LSTPRM, PRMBUF, PRMEND, CFSPND
                        .GLOBL  CFHOLD, LOGDVU, LOGBAS
                        .GLOBL  LCOL, $QTSET, $TECO, CD$TOP, LOGCHK
                        .GLOBL  $WILD, ERRSEV, UERSEV, PASLIN, LOGBAS, LOGDVU
                        .GLOBL  LSTPL, SDCB, SDCBND, VQUAN0, VQUAN3
                        .GLOBL  VQUAN1, VQUN1A, VQUAN2, VHIPCT, VQUAN0, VQUAN3
                        .GLOBL  DCTRD, DCCRD, DCTWR, DCCWR, ASNSRC
                        .GLOBL  VCORTM, KMPRMT, MXPRMT
                        .GLOBL  RDB, RDBEND, RT$NAM, RT$$SZ, CLDEVX, SDDVU
                        .GLOBL  SDCBSZ, LSTSL, LSTATE
                        .GLOBL  TK1VAL, CINDAT, SYSDAT, SYTIMH, SYTIML
                        .GLOBL  BASMAP, LOMAP, HIMAP, JCXPGS
                        .GLOBL  SMRSIZ, SRTSIZ, CSHSIZ, TK1SEC
                        .GLOBL  TSXLN, TSXSIT, GRT1, TRGRET, LICTXT, SUPCOD, NAMTOP
                        .GLOBL  SYINDX, SYUNIT, NUMDEV, PNAME
                        .GLOBL  OF$DEV, OF$UNT, OF$FIL, OF$FLG, SYNAME
                        .GLOBL  OF$$SZ, OT$RON, RESDEV, $TAPE
                        .GLOBL  KMNBAS, ODTBAS, $CTRLD
                        .GLOBL  LSW6, $SNWTT, PF$SYS, PF$IOW, $DEBUG
                        .GLOBL  $INDDF, $INDRN, IN$ACT, IN$CNT, IN$CMD, INDSAV
                        .GLOBL  $PHONE, INVEC, LMXLN, MXVEC, $INIT, $DEAD, $HARD
                        .GLOBL  ITRMTP, LMXPRM, LSW7, CFSTS, CF$IND, CF$QUT
                        .GLOBL  CFABLV, MONVEC, LBSPRI, MAXPRI, MXJPRI, LPRI
                        .GLOBL  LOGCHN, LOGFLG, LOGPTR, LOGBUF, LOGBLK
                        .GLOBL  LF$OPN, LF$WRT, UCLBLK, UCLDAT
                        .GLOBL  CSHHD, UC$NDC, UC$MDC, CVTUC
                        .GLOBL  CMDBUF, PAUMSG, RDCMD, DKSAV, SYSAV, CVTTAB, SEARCH
                        .GLOBL  INVOPT, FKILL, ABRTCF, INDABT, 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, 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, HANIDX, HNBUF
                        .GLOBL  ACROCT, HANBSY, CSIMS1, MISSEQ, NOIND, POPCF
                        .GLOBL  BADPMT, BADPRI, TOTXT, CRLF, HIPRI, STLGHD, LOGCLS, R50LOG
                        .GLOBL  BDLGOP, SPLHLA, LDOPHD, PRTFIX, PRTSPC
                        .GLOBL  DLTXT, OCTFIX, PRTTTP, NATXT, NOTXT, YESTXT, NINTXT
                        .GLOBL  PRTUNM, SYHD1, SYHD2, PRTLN, DETTXT, RNMS
                        .GLOBL  SWPTX, LOCKTX, PRTDC3, KBMSG, DIVIDE, PRTDC2
                        .GLOBL  COLOO, CPUAH, CPUAL, PRTTMV, NOFIL, CMDBUF, CALUCL
                        .GLOBL  NOUDC, DEVHD1, ASNHD1, ASNHD2, SHMTH1, SHMTH2, PRTTMD
                        .GLOBL  CVDVNM, 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, NUMTB1, MSGEND, NOTON, GAGMSG
                        .GLOBL  YELEMT, LINFRE, DJABMS, DLMSG, INVTIM, DMTALL
                        .GLOBL  SHTMSG, AUTHFN, SPLACT, DOSTOP, OFFEMT, KILEMT, UPTMMS
                        .GLOBL  TMTOTH, DIVSOR, TMTOTL, PRTPCT
```

```
172                                     .GLOBL  OTHRON,SPLPND,STPASK,SRTSMS
173                                     .GLOBL  SIZEMT,ASNOVF,INVLDM,CSIMS4,MNTARG,HUPARG,R50TT
174                                     .GLOBL  KMNNAM,CCLNAM,OTRMNT,CHKDEV,DMTSUB,CMDCCL
175                                     .GLOBL  SHOHD,SUBTXT,MNTTXT,SRTTXT,TOTMMS,UMSSMS,SSRMAP
176                                     .GLOBL  TSXSMS,USRMMS,JCXSMS,DZTXT,OCTPRT,SJEMT,RJEMT
177                                     .GLOBL  PRTR50,PRTDAT,PRTTOD,PRTTIM,INVDEV,ALFN,R50DK
178                                     .GLOBL  DETHD,DETARG,RUNMS,NOFRDL,R50MON,INVDAT,MUL32,COAF
179                                     .GLOBL  AR$PRJ,AR$PRG,AR$CON,AR$CNT,AR$CPH,AR$CPL,AR$UNM
180                                     .GLOBL  AR$DMY,AR$$SZ,ARNRPB,$SLON,$SLTTY,$SLLET
181                                     .GLOBL  PRTWRN,SLMXLN,VLDSYS,$LOFCF,CSHMSG,CCSPRV
182                                     .GLOBL  INSTBL,INSTBN,AF$NOW,AF$HIE,AF$NOI,$NOINT,II$$SZ
183                                     .GLOBL  AF$IOP,$RNIOP,$RNMLK,II$NAM,II$FLG,II$PRV,II$NPV
184                                     .GLOBL  NMSGBF,$NOABT
```

```
                        ;
                        ;  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
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  CMDV2A
                        NAME'NA: .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  NAME2A
                        L     =       .
                              .ASCIZ  /STRNG/
                              .CSECT  CMDV2A
                              .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  CMDV2A
                                .WORD   0
                                .CSECT  TSKM2A
                                .ENDM   TBLEND
```

```
                                        .SBTTL  Data areas
                                ;-------------------------------------------------------------------
                                ;    Data areas:
                                ;
000000  000000                  SQZDEV: .WORD   0               ;Name of device being squeezed
000002  000000                  ALC1DV: .WORD   0               ;Name of 1st device on allocation list
000004  073634                  R50SET: .RAD50  /SET/
000006  102700                  R50UP:  .RAD50  /UP /
000010                          REMRDB: .BLKW   5
000022  003344                  R50ADD: .RAD50  /ADD/
000024  014724  021145          R50DEL: .RAD50  /DELETE/
000030  070525                  R50REM: .RAD50  /REM/
000032  000000                  IATTRB: .WORD   0
                                ;
                                ;  Byte data
                                ;
000034      000                 DMTNLG: .BYTE   0
000035      000                 OFFNWF: .BYTE   0
000036      000                 YELFLG: .BYTE   0
                                        .EVEN
                                ;
                                ;  Emt to delete a spool file
                                ;
000040      000     126         DELEMT: .BYTE   0,126
000042  000007                          .WORD   7
000044  000000                          .WORD   0               ;File ID goes here
                                ;
                                ;  Emt to set spool HOLD mode
                                ;
000046      000     126         SPHLEM: .BYTE   0,126
000050  000015                          .WORD   15
000052  000000                          .WORD   0               ;Address of SDCB
                                ;
                                ;  Emt to set spool NOHOLD mode
                                ;
000054      000     126         SPNHEM: .BYTE   0,126
000056  000016                          .WORD   16
000060  000000                          .WORD   0               ;Address of SDCB
                                ;
                                ;  Time to wait for check spooler activity.
                                ;
000062  000000  000170          TIMSPL: .WORD   0,60.*2.        ;2 second timer
```

```
                                        .SBTTL  Tables of commands and options
                                ;-----------------------------------------------------------
                                ;  Define option switches for the INSTALL ADD command
                                ;
000066                                  TBLDEF  INS,2
000002                                  CMDDEF  BYPASN,INSATR,AF$BYA
000010                                  CMDDEF  DUP,INSATR,AF$DUP
000016                                  CMDDEF  DEB*UG,INSATR,AF$DBG
000024                                  CMDDEF  HI*GH,INSATR,AF$HIE
000032                                  CMDDEF  IND,INSATR,AF$IND
000040                                  CMDDEF  IO*MAP,INSATR,AF$IOP
000046                                  CMDDEF  IO*PAGE,INSATR,AF$IOP
000054                                  CMDDEF  LOCK,INSATR,AF$PLK
000062                                  CMDDEF  MEM*LOCK,INSATR,AF$MEM
000070                                  CMDDEF  NONI*NTERACTIVE,INSATR,AF$NOI
000076                                  CMDDEF  NOWA*IT,INSATR,AF$NOW
000104                                  CMDDEF  NOWI*NDOW,INSATR,AF$NPW
000112                                  CMDDEF  PRIV*ILEGED,PRVOPT,0
000120                                  CMDDEF  SC*CA,INSATR,AF$CCA
000126                                  CMDDEF  SETUP,INSATR,AF$SET
000134                                  CMDDEF  SING*LECHAR,INSATR,AF$SCA
000142                                  CMDDEF  T*RANSPARENT,INSATR,AF$TPO
000150                                  CMDDEF  TSXUCL,INSATR,AF$UCL
000156                                  TBLEND

                                ;-----------------------------------------------------------
                                ;  Define option switches for the MOUNT command
                                ;
000066                                  TBLDEF  MTOP,1
000162                                  CMDDEF  WR*ITE,0
000166                                  CMDDEF  NOW*RITE,1
000172                                  TBLEND

                                ;-----------------------------------------------------------
                                ;  Define option switches for the DISMOUNT command
                                ;
000066                                  TBLDEF  DMTQ,1
000176                                  CMDDEF  LOG,DMTLGQ
000202                                  CMDDEF  NOLOG,DMTNLQ
000206                                  CMDDEF  WARN,DMTLGQ
000212                                  CMDDEF  NOWARN,DMTNLQ
000216                                  TBLEND

                                ;-----------------------------------------------------------
                                ;  Define option switches for DETACH command
                                ;
000066                                  TBLDEF  DET,1
000222                                  CMDDEF  CH*ECK,DETCHK
000226                                  CMDDEF  KI*LL,DETKIL
000232                                  TBLEND

                                ;-----------------------------------------------------------
                                ;  Options for the MONITOR command.
                                ;
000066                                  TBLDEF  MON,1
000236                                  CMDDEF  IO*WAIT,PF$IOW
000242                                  CMDDEF  SY*STEM,PF$SYS
```

```
      58 000246                                 TBLEND
      59
      60                                ;-------------------------------------------------------------------------------
      61                                ;  Options for the OFF (BYE, KJOB, LOGOFF) commands.
      62                                ;
      63 000066                                 TBLDEF  OFF,1
      64 000252                                 CMDDEF  N*OWARN,OFF-NWN
      65 000256                                 TBLEND
```

```
                                ;-----------------------------------------------------------------
                                ;  Define options for the SPOOL command
                                ;  Entry 1 = Option name
                                ;  Entry 2 = Processing routine
                                ;
000066                                TBLDEF  SPL,1
000262                                CMDDEF  A*LIGN,SPLALN
000266                                CMDDEF  B*ACKUP,SPLBAK
000272                                CMDDEF  D*ELETE,SPLDEL
000276                                CMDDEF  F*ORM,SPLFRM
000302                                CMDDEF  H*OLD,SPLHLD
000306                                CMDDEF  NOH*OLD,SPLNHL
000312                                CMDDEF  L*OCK,SPLLK
000316                                CMDDEF  M*ULTIPLE,SPLMUL
000322                                CMDDEF  SK*IP,SPLSKP
000326                                CMDDEF  SI*NGLE,SPLSNG
000332                                CMDDEF  ST*ATUS,SPLSTA
000336                                TBLEND
```

```
                                        .SBTTL  PAUSE command
                                ;-------------------------------------------------------------------------
                                ;  THE PAUSE COMMAND IS USED TO SUSPEND EXECUTION TEMPORARILY
                                ;  WHILE PROCESSING A COMMAND FILE.
                                ;  WHEN THE PAUSE COMMAND IS FOUND EXECUTION IS SUSPENDED UNTIL
                                ;  THE USER ENTERS A CARRIAGE RETURN FROM THE KEYBOARD.
                                ;
000066  016767  000000G 000000G CMDPAU: MOV     CFPNT,CFSPND    ;SUSPEND COMMAND FILE
000074  001441                         BEQ     9$              ;BR IF COMMAND FILE NOT RUNNING
000076  004767  000000G                 CALL    CFSTOP          ;Suspend file
                                ;
                                ;  Set command-file-abort flag so that command file execution will
                                ;  be aborted if control-C is typed in response to the pause.
                                ;
000102  032761  000000G 000000G         BIT     #$SCCA,LSW5(R1) ;Is control-C abort override in effect?
000110  001003                         BNE     3$              ;Br if yes -- Don't abort on control-C
000112  052761  000000G 000000G         BIS     #$CFABT,LSW6(R1);Abort command files if ctrl-C typed
                                ;
                                ;  If TTY QUIET is set, print pause message for operator
                                ;
000120  032761  000000G 000000G 3$:     BIT     #$QUIET,LSW4(R1);IS QUIET SET ON?
000126  001403                         BEQ     2$              ;BR IF NOT
000130                                  .PRINT  #CMDBUF         ;DISPLAY THE PAUSE COMMAND
                                ;
                                ;  Print pause message and wait for response
                                ;
000136                          2$:     .PRINT  #PAUMSG         ;PRINT PAUSE MESSAGE
000144                          1$:     .TTYIN                  ;ACCEPT A LINE OF INPUT
000150  120027  000012                  CMPB    R0,#LF
000154  001373                         BNE     1$
                                ;
                                ;  Restart the command file
                                ;
000156  042761  000000G 000000G         BIC     #$CFABT,LSW6(R1);Clear command-file abort flag
000164  016700  000000G                 MOV     CFSPND,R0       ;RESTART COMMAND FILE
000170  004767  000000G                 CALL    CFSTRT
000174  005067  000000G                 CLR     CFSPND
000200  000167  000000G         9$:     JMP     RDCMD

                                        .SBTTL  DISPLAY command
                                ;-------------------------------------------------------------------------
                                ;  Display a line from within a command file.
                                ;
000204  124327  000040          CMDDSP: CMPB    -(R3),#' ;BACKUP OVER LEADING SPACES
000210  001775                         BEQ     CMDDSP
000212  005203                         INC     R3              ;POINT TO FIRST SEPARATOR
000214  122327  000040                 CMPB    (R3)+,#' ;IS IT A SPACE?
000220  001401                         BEQ     1$              ;BR IF YES (SKIP IT)
000222  005303                         DEC     R3              ;POINT TO 1ST SEPARATOR
000224                          1$:     .PRINT  R3              ;PRINT THE ARGUMENT OF THE DISPLAY COMMAND
000230  000167  000000G                 JMP     RDCMD
```

```
                                        .SBTTL  ASSIGN command
                                ;-------------------------------------------------------
                                ;  PROCESS THE 'ASSIGN' COMMAND.
                                ;
                                ;  SEE IF FULL FILE ID IS SPECIFIED OR JUST DEVICE NAME.
                                ;
000234  004767  0000000        CMDASN: CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACES
000240  105067  0000000                CLRB    ASKLNM          ;SET FLAG FOR LONG ASSIGN
000244  126227  177777  000072        CMPB    -1(R2),#':      ;DOES LOGICAL NAME HAVE COLON?
000252  001001                        BNE     5$              ;BR IF NOT
000254  105042                        CLRB    -(R2)           ;CLEAR IT FOR NOW
000256  010201                5$:     MOV     R2,R1           ;POINT TO NULL AT END OF LINE
000260  020103                1$:     CMP     R1,R3           ;SCAN BACK TOWARDS FRONT
000262  101002                        BHI     2$              ;BRANCH IF MORE TO DO
000264  000167  0000000               JMP     BADCMD          ;MUST HIT SOME DELIM
000270  114100                2$:     MOVB    -(R1),R0        ;GET NEXT CHAR BACK
000272  004767  000320                CALL    ASDELM          ;IS IT A DELIMITER?
000276  001370                        BNE     1$              ;IF NOT KEEP SCANNING
                                ;  WE SCANNED BACK TO A DELIMITER -- SEE WHERE IT IS
                                ;  IN COMMAND.
000300  120027  000072                CMPB    R0,#':          ;REPLACE ':' AND ' ' WITH '='
000304  001414                        BEQ     3$
000306  120027  000040                CMPB    R0,#' 
000312  001013                        BNE     4$
000314  020103                6$:     CMP     R1,R3           ;HAVE WE REACHED FRONT OF COMMAND LINE?
000316  101407                        BLOS    3$              ;BR IF YES
000320  124127  000040                CMPB    -(R1),#' '      ;SKIP OVER ALL CONSECUTIVE BLANKS
000324  001773                        BEQ     6$
000326  121127  000072                CMPB    (R1),#':        ;ALLOW "ASSIGN DX1:  DK" CONSTRUCT
000332  001401                        BEQ     3$
000334  005201                        INC     R1              ;POINT TO 1ST BLANK CHARACTER
000336  112711  000075        3$:     MOVB    #'=,(R1)
000342  010100                4$:     MOV     R1,R0           ;HOW MANY CHARS IN NAME TO
000344  160300                        SUB     R3,R0           ;LEFT OF DELIMITER?
000346  020027  000003                CMP     R0,#3           ;DEVICE NAME <= 3 CHARS
000352  003021                        BGT     ASFID           ;LONGER==>FULL FILE NAME.
                                ;
                                ;  SIMPLE ASSIGN OF PHYSICAL TO LOGICAL DEVICE.
                                ;  REFORMAT COMMAND TO 'DEV:A=DEV:' SO CSISPC CAN
                                ;  HANDLE IT.
                                ;
000354  112722  000072        ASSMPL: MOVB    #':,(R2)+       ;PUT COLON AT END OF 2ND NAME
000360  105022                        CLRB    (R2)+           ;NULL TO TERMINATE STRING
000362  010204                        MOV     R2,R4           ;MAKE ROOM AT END OF 1ST NAME
000364  062704  000002                ADD     #2,R4           ;BY SLIDING CHARS OVER 2 PLACES
000370  020201                1$:     CMP     R2,R1           ;WORK BACK DOWN TO '='
000372  001402                        BEQ     2$              ;BRANCH AFTER '=' MOVED
000374  114244                        MOVB    -(R2),-(R4)     ;SLIDE OVER TO RIGHT
000376  000774                        BR      1$
000400  112722  000072        2$:     MOVB    #':,(R2)+       ;PUT COLON AFTER 1ST DEVICE NAME
000404  112712  000101                MOVB    #'A,(R2)        ;PUT IN DUMMY FILE NAME 'A'.
000410  105267  0000000               INCB    ASKLNM          ;REMEMBER TO KILL FILE NAME LATER
000414  000403                        BR      ASCIS           ;GO DO .CSISPC
                                ;
                                ;  REQUEST ASSIGNMENT OF LOGICAL DEVICE TO NAMED FILE.
                                ;
000416  112722  000072        ASFID:  MOVB    #':,(R2)+       ;PUT COLON AT END OF LOGICAL NAME
```

```
      58 000422  105012                          CLRB    (R2)            ;NULL TO END STRING
      59                                 ;
      60                                 ;  CALL CSI IN SPECIAL MODE TO PARSE ASSIGN COMMAND.
      61                                 ;
      62 000424  010605                  ASCIS:  MOV     SP,R5           ;SAVE STACK POINTER
      63 000426                                  .CSISPC #KCSIBF,#ASDEX,R3
      64 000442  010506                          MOV     R5,SP           ;RESET TOP OF STACK
      65 000444  103010                          BCC     ASCSOK          ;BRANCH IF .CSISPC OK
      66                                 ;  ERROR OCCURED ON .CSISPC
      67 000446  113705  0000000         CSIERR: MOVB    @#ERRLOC,R5     ;GET ERROR CODE
      68 000452  006305                          ASL     R5              ;CONVERT TO WORD INDEX
      69 000454  016504  0000000                 MOV     KCSIMS(R5),R4   ;GET ADDR OF ERROR MESSAGE
      70 000460                                  FABORT  R4              ;PRINT ERROR MESSAGE
      71                                 ;
      72                                 ;  Move the assign information into the assign table.
      73                                 ;  The physical device (and possibly file name)
      74                                 ;  description is in KCSIBF in the area for chan #0.
      75                                 ;  The logical device block is in the space for chan #3.
      76                                 ;
      77                                 ;  If "physical" device name used in assign statement is actually
      78                                 ;  a logical name (previously assigned), convert it to the corresponding
      79                                 ;  physical device name.
      80                                 ;
      81 000466  016700  0000000         ASCSOK: MOV     KCSIBF,R0       ;GET PHYSICAL DEVICE NAME FOR ASSIGN
      82 000472  004767  0000000                 CALL    ASNSRC          ;SEE IF IT IS ACTUALLY A LOGICAL DEV NAME
      83 000476  103403                          BCS     3$              ;BR IF NOT
      84 000500  016267  0000000 0000000        MOV     AT$DEV(R2),KCSIBF;REPLACE LOGICAL NAME WITH REAL PHYSICAL NAME
      85 000506  105767  0000000         3$:     TSTB    ASKLNM          ;MUST WE KILL FILE NAME?
      86 000512  001402                          BEQ     1$              ;BRANCH IF NOT
      87 000514  005067  0000020                 CLR     <KCSIBF+2>      ;KILL THE NAME
      88                                 ;  Clear any previous assign with this logical name.
      89 000520  016700  0000360         1$:     MOV     KCSIBF+30.,R0   ;GET LOGICAL DEVICE NAME
      90 000524  004767  0000000                 CALL    ASNSRC          ;SEE IF NAME IS ALREADY ASSIGNED
      91 000530  103010                          BCC     2$              ;BR IF ASSIGN ALREADY EXISTS (REUSE THIS ENTRY
      92                                 ;  Search for a free assign block.
      93 000532  005000                          CLR     R0              ;SEARCH FOR FREE ASSIGN BLOCK
      94 000534  004767  0000000                 CALL    ASNSRC
      95 000540  103004                          BCC     2$              ;BR IF FOUND ONE
      96                                 ;  ASSIGN TABLE IS FULL.
      97 000542                                  FABORT  #ASNOVF
      98                                 ;
      99                                 ;  FOUND A FREE SLOT (POINTED TO BY R2) MOVE ASSIGN
     100                                 ;  INFO INTO BLOCK.
     101                                 ;
     102 000552  016762  0000360 0000000 2$:     MOV     KCSIBF+30.,AT$LOG(R2);MOVE IN LOGICAL DEVICE NAME
     103 000560  016762  0000100 0000000         MOV     KCSIBF+8.,AT$SIZ(R2);MOVE IN FILE LENGTH
     104 000566  012703  0000000                 MOV     #KCSIBF,R3
     105 000572  012362  0000000                 MOV     (R3)+,AT$DEV(R2);MOVE IN REAL DEVICE NAME
     106 000576  012362  0000000                 MOV     (R3)+,AT$FIL(R2);MOVE IN FILE NAME
     107 000602  012362  0000020                 MOV     (R3)+,AT$FIL+2(R2)
     108 000606  011362  0000000                 MOV     (R3),AT$EXT(R2) ;MOVE IN FILE EXTENSION
     109 000612  000167  0000000                 JMP     RDCMD
```

```
                                ;-----------------------------------------------------------
                                ;  ASDELM IS CALLED TO CHECK IF THE CHARACTER IN R0
                                ;  IS ONE OF THE CHARACTERS ':', '=', OR ' '.
                                ;  ON RETURN THE CONDITION CODE IS SET FOR BEQ/BNE.
                                ;  ALL REGISTERS ARE PRESERVED.
                                ;
000616  120027  000075        ASDELM: CMPB    R0,#'=          ;CHECK EQUAL SIGN
000622  001405                        BEQ     1$
000624  120027  000072                CMPB    R0,#':          ;CHECK COLON
000630  001402                        BEQ     1$
000632  120027  000040                CMPB    R0,#'           ;CHECK SPACE
000636  000207                1$:     RETURN          ;RETURN WITH CC SET.
```

```
                                        .SBTTL  DEASSIGN command
                                ;-------------------------------------------------------------
                                ;  PROCESS THE DEASSIGN COMMAND
                                ;
000640  004767  0000000  CMDDSN:  CALL    CVTTAB          ;CONVERT TAB AND FF TO SPACES
000644  105713                   TSTB    (R3)            ;DEASSIGN ALL OR ONE?
000646  001010                   BNE     DEAS1           ;BR TO DEASSIGN ONE NAME
                                ;
                                ;  REQUEST TO DEASSIGN ALL LOGICAL NAMES
                                ;  (SIMPLY ZERO THE ASSIGN TABLE)
                                ;
000650  012702  0000000          MOV     #ASNTBL,R2      ;POINT TO START OF TABLE
000654  005022                1$:       CLR     (R2)+
000656  020227  0000000          CMP     R2,#ASNEND      ;DONE ALL?
000662  103774                   BLO     1$              ;BR IF NOT
000664  000167  0000000  DAJMP:   JMP     RDCMD
                                ;
                                ;  REQUEST TO DEASSIGN A PARTICULAR LOGICAL DEVICE NAME
                                ;
000670  004767  0000000  DEAS1:   CALL    GTRD50          ;ACCRUE THE DEVICE NAME
000674  016700  0000000          MOV     R50BUF,R0       ;PICK UP DEVICE NAME
000700  012702  0000000          MOV     #ASNTBL,R2      ;POINT TO START OF ASSIGN TABLE
000704  020062  0000000  1$:      CMP     R0,AT$LOG(R2)   ;IS THIS ENTRY FOR THE DEVICE?
000710  001004                   BNE     2$              ;BR IF NOT
000712  005062  0000000          CLR     AT$LOG(R2)      ;DEASSIGN THE LOGICAL NAME
000716  005062  0000000          CLR     AT$DEV(R2)      ;CLEAR PHYSICAL DEVICE NAME ALSO
000722  062702  0000000  2$:      ADD     #AT$$SZ,R2      ;POINT TO NEXT ASSIGN TABLE ENTRY
000726  020227  0000000          CMP     R2,#ASNEND      ;REACHED END OF TABLE?
000732  103764                   BLO     1$              ;BR IF MORE TO DO
000734  000753                   BR      DAJMP
```

```
                                        .SBTTL  ALLOCATE command
                                ;-----------------------------------------------------------------------
                                ;  Allocate a device for exclusive access by this job.
                                ;  The form of this command is:
                                ;    ALLOCATE dev[:],... logical_name
                                ;
000736                          CMDALC:
                                ;
                                ;  If no devices were specified, treat this like a SHOW ALLOCATE command
                                ;
000736  005067  177040                CLR     ALC1DV          ;Say no allocat device seen yet
000742  004767  0000000               CALL    CVTTAB          ;Convert tab and FF chars to spaces
000746  004767  0000000               CALL    SKPSPC          ;Skip over spaces
000752  105713                        TSTB    (R3)            ;Any devices specified?
000754  001002                        BNE     1$              ;Br if yes
000756  000167  0000000               JMP     SOPALC          ;Do SHOW ALLOCATE command
                                ;
                                ;  See if we are authorized to allocate devices
                                ;
000762  032767  0000000 0000000 1$:   BIT     #PO$ALC,PRIVCO  ;Are we authorized to allocate devices?
000770  001004                        BNE     10$             ;Br if yes
000772                                FABORT  #EM$ALC         ;Not authorized for ALLOCATE
                                ;
                                ;  Begin loop to get the name of each device to be allocated
                                ;
001002  004767  0000000         10$:  CALL    SKPSPC          ;Skip over any spaces
                                ;
                                ;  Accrue the next device name
                                ;
001006  004767  0000000               CALL    GTRD50          ;Accrue the device name
001012  122327  000072                CMPB    (R3)+,#':       ;Was a colon specified with the device name?
001016  001401                        BEQ     6$              ;Br if yes
001020  005303                        DEC     R3              ;Backup pointer
001022  010346                6$:     MOV     R3,-(SP)        ;Save command pointer
001024  004767  0000000               CALL    SKPDLM          ;Skip legal delimiters (blank(s), comma, EOL)
001030  012603                        MOV     (SP)+,R3        ;Restore command pointer
001032  103435                        BCS     41$             ;Invalid character following name
001034  016702  0000000               MOV     R50BUF,R2       ;Get the name of the device
001040  010267  0000000               MOV     R2,ALCDEV       ;Set name of device being allocated
                                ;
                                ;  Save the first device name
                                ;
001044  005767  176732                TST     ALC1DV          ;Have we seen the first device yet?
001050  001002                        BNE     2$              ;Br if yes
001052  010267  176724                MOV     R2,ALC1DV       ;Save the name of the 1st device
                                ;
                                ;  Do the allocation for this device
                                ;
001056  012700  0000000         2$:   MOV     #ALDEMT,R0      ;Point to EMT arg block to do the allocation
001062  104375                        EMT     375             ;Try to allocate the device
001064  103051                        BCC     3$              ;Br if allocation was successful
001066  010005                        MOV     R0,R5           ;Save returned error value
                                ;
                                ;  An error occurred on the allocation.  Print error message.
                                ;
001070  113702  0000000               MOVB    @#ERRLOC,R2     ;Get allocation error code
001074  120227  000001                CMPB    R2,#1           ;Device already allocated by someone else?
```

```
                        BNE     4$              ;Br if not
                        FERR    #EM$DAA         ;Device allocated by another job
                        BR      7$
4$:                     CMPB    R2,#2           ;Invalid device being allocated?
                        BNE     5$              ;Br if not
41$:                    FABORT  #EM$IAD         ;Invalid device for allocation
5$:                     CMPB    R2,#3           ;Allocation table full?
                        BNE     8$              ;Br if not
                        FABORT  #EM$ATF         ;Allocation table full
8$:                     CMPB    R2,#4           ;Device in use by another user?
                        BNE     3$              ;Br if not
                        FERR    #EM$DIU         ;Device is in use by another user
7$:                     CALL    PRTDEC          ;Print the number of the job that has the dev
                        .PRINT  #CRLF           ;End of line
                ;
                ;  See if there are more devices to be allocated
                ;
3$:                     CALL    SKPSPC          ;Skip over any spaces
                        CMPB    (R3)+,#',       ;Is there another device to allocate?
                        BEQ     10$             ;Loop if yes
                ;
                ;  We have finished allocating all devices.
                ;  See if we need to assign a logical name to the first device we allocated.
                ;
                        TSTB    -(R3)           ;Was a logical device name specified?
                        BEQ     9$              ;Br if not
                        CALL    GTRD50          ;Accrue the logical device name
                        MOV     R50BUF,R5       ;Get the logical device name
                        MOV     ALC1DV,R0       ;Get the physical device name
                        CALL    DOASGN          ;Do the assignment
                ;
                ;  Finished
                ;
9$:                     JMP     RDCMD           ;Finished with command
```

```
                                        .SBTTL  DEALLOCATE command
                                ;-------------------------------------------------------------------------------
                                ;  Deallocate an allocated device.
                                ;
001252  005067  176524          CMDDLC: CLR     ALC1DV          ;Clear deallocated device counter
001256  032767  000000G 000000G        BIT     #PO$ALC,PRIVCO  ;Are we authorized to allocate devices?
001264  001004                        BNE     1$              ;Br if yes
001266                                  FABORT  #EM$ALC         ;Not authorized to deallocate
                                ;
                                ;  See if /ALL was specified
                                ;
001276  004767  0000000         1$:     CALL    CVTTAB          ;Convert tab and FF chars to spaces
001302  004767  0000000                 CALL    CVTUC           ;Convert lower case chars to upper case
001306  004767  0000000                 CALL    SKPSPC          ;Skip over spaces
                                ;
                                ;  Find devices for deallocation.
                                ;
001312  111300                  2$:     MOVB    (R3),R0         ;Get 1st char of operand
001314  001473                          BEQ     10$             ;No devices specified, deallocate all
001316  120027  000057          3$:     CMPB    R0,#'/          ;Was a switch specified?
001322  001012                          BNE     4$              ;Br if not
001324  005203                          INC     R3              ;Skip over slash
001326  004767  0000000                 CALL    SKPSPC          ;Skip any spaces
001332  122327  000101                  CMPB    (R3)+,#'A       ;"ALL"?
001336  001465                          BEQ     12$             ;Br if yes
001340                                  FABORT  #INVOPT         ;Invalid option
                                ;
                                ;  Deallocate a list of devices
                                ;  Accrue the next device name
                                ;
001350  005267  176426          4$:     INC     ALC1DV          ;Count deallocated devices
001354  004767  0000000                 CALL    SKPSPC          ;Skip over any spaces
001360  004767  0000000                 CALL    GTRD50          ;Accrue the device name
001364  122327  000072                  CMPB    (R3)+,#':       ;Was colon specified following name?
001370  001401                          BEQ     5$              ;Br if yes
001372  005303                          DEC     R3              ;Backup pointer
001374  004767  0000000         5$:     CALL    SKPDLM          ;Skip valid delimiters (blank(s), comma)
001400  103435                          BCS     7$              ;Br if invalid character
                                ;
                                ;  Deallocate the device specified.
                                ;
001402  016702  0000000                 MOV     R50BUF,R2       ;Get the name of the device
001406  010267  0000000                 MOV     R2,ALCDEV       ;Set name of device being allocated
001412  012700  0000000                 MOV     #DLCEMT,R0      ;Point to EMT argument block
001416  104375                          EMT     375             ;Do the deallocation
001420  103334                          BCC     2$              ;Br if ok
001422  010005                          MOV     R0,R5           ;Save job # if owned by someone else
001424  113700  0000000                 MOVB    @#ERRLOC,R0     ;Get error code
001430  120227  000001                  CMPB    R2,#1           ;Device allocated by someone else?
001434  001014                          BNE     6$              ;Br if not
001436                                  FERR    #EM$DAA         ;Device allocated by another job
001452  004767  0000000                 CALL    PRTDEC          ;Print the number of the job that has the dev
001456                                  .PRINT  #CRLF           ;End of line
001464  000712                          BR      2$
001466  120227  000002          6$:     CMPB    R2,#2           ;Invalid device being allocated?
001472  001307                          BNE     2$              ;Br if not
001474                          7$:     FABORT  #EM$IAD         ;Invalid device for allocation
```

```
                                ;
                                ;  Deallocate all devices allocated by this user
                                ;
001504  005767  176272          10$:    TST     ALC1DV          ;Were any devices specified?
001510  001006                          BNE     20$             ;Yes, end of command
001512  005067  0000000         12$:    CLR     ALCDEV          ;Say to deallocate all devices
001516  012700  0000000                 MOV     #DLCEMT,R0      ;Point to EMT argument block
001522  104375                          EMT     375             ;Do the deallocation
001524  000400                          BR      20$
                                ;
                                ;  Finished
                                ;
001526  000167  0000000         20$:    JMP     RDCMD           ;Finished with command
```

```
                                        .SBTTL  REMOVE command
                                ;-----------------------------------------------------------------------------
                                ;  The REMOVE command has two functions:
                                ;  (1) To remove a device handler from memory.  This function is a NOP
                                ;      under TSX-Plus;
                                ;  (2) To remove a named PLAS region.
                                ;
001532  004767  000000G         CMDREM: CALL    CVTTAB          ;Convert tab and FF chars to spaces
                                ;
                                ;  Quit if we have reached the end of the command line
                                ;
001536  004767  000000G         1$:     CALL    SKPSPC          ;Skip past leading spaces
001542  105713                          TSTB    (R3)            ;Are we at the end of the command?
001544  001436                          BEQ     9$              ;Br if yes
                                ;
                                ;  Accrue the name of a device or region
                                ;
001546  004767  000000G                 CALL    GTRD50          ;Accrue device or region name
                                ;
                                ;  If the name is longer than 3 characters then it must be a region
                                ;  name; otherwise check to see if it is a device name.
                                ;
001552  005767  000020                  TST     R50BUF+2        ;Is name longer than 3 characters?
001556  001012                          BNE     2$              ;Br if yes
001560  121327  000072                  CMPB    (R3),#':        ;Is name terminated with colon?
001564  001002                          BNE     3$              ;Br if not
001566  105723                          TSTB    (R3)+           ;Skip over colon
001570  000407                          BR      5$              ;Must have been a device name
001572  016705  000000G         3$:     MOV     R50BUF,R5       ;Get 1st 3 chars of name
001576  004767  000000G                 CALL    CHKDEV          ;See if this is the name of a device
001602  103002                          BCC     5$              ;Br if device name
                                ;
                                ;  This is a request to remove a named PLAS region.
                                ;
001604  004767  000036          2$:     CALL    REMRGN          ;Remove a region
                                ;
                                ;  Skip over any separating comma
                                ;
001610  122327  000054          5$:     CMPB    (R3)+,#',       ;Is there a separating comma?
001614  001401                          BEQ     4$              ;Br if yes
001616  005303                          DEC     R3              ;Point back to skipped char
001620  105713                  4$:     TSTB    (R3)            ;Reached end of command?
001622  001407                          BEQ     9$              ;Br if yes
001624  121327  000040                  CMPB    (R3),#' '       ;Space separator?
001630  001742                          BEQ     1$              ;Br if yes
001632                                  FABORT  #EM$CSE         ;Command syntax error
                                ;
                                ;  Finished
                                ;
001642  000167  000000G         9$:     JMP     RDCMD
```

```
                                ;-------------------------------------------------------------------------------
                                ;  Eliminate a named PLAS region.
                                ;
                                ;  Inputs:
                                ;   R50BUF = Rad50 name of the region.
                                ;
001646  010246                  REMRGN: MOV     R2,-(SP)
                                ;
                                ;  First attempt to attach to a private region with that name.
                                ;
001650  012702  000010'                 MOV     #REMRDB,R2      ;Point to region definition block
001654  005062  000000G                 CLR     R.GSIZ(R2)      ;Clear size word
001660  012762  000000C 000000G         MOV     #<RS.GBL!RS.PVT>,R.GSTS(R2) ;Function = Attach private region
001666  016762  000000G 000000G         MOV     R50BUF,R.NAME(R2) ;Set name of region
001674  016762  000002G 000002G         MOV     R50BUF+2,R.NAME+2(R2)
001702                                  .CRRG   #XAREA,R2       ;Attempt to attach to region
001720  032762  000000G 000000G         BIT     #RS.CRR,R.GSTS(R2) ;Were we able to attach?
001726  001025                          BNE     1$              ;Br if yes
                                ;
                                ;  Unable to attach to private region.
                                ;  Try to attach to public region.
                                ;
001730  012762  000000G 000000G         MOV     #RS.GBL,R.GSTS(R2) ;Function = Attach global region
001736                                  .CRRG   #XAREA,R2       ;Attempt to attach to it
001754  032762  000000G 000000G         BIT     #RS.CRR,R.GSTS(R2) ;Was attachment successful?
001762  001007                          BNE     1$              ;Br if yes
                                ;
                                ;  Error -- We are unable to attach to region
                                ;
001764                                  FERR    #EM$UAR         ;Unable to attach to region
002000  000421                          BR      9$
                                ;
                                ;  We successfully attached to the region.
                                ;  Try to eliminate the region.
                                ;
002002  012762  000000G 000000G 1$:     MOV     #RS.EGR,R.GSTS(R2) ;Function = Eliminate region
002010                                  .ELRG   #XAREA,R2       ;Try to eliminate the region
002026  103006                          BCC     9$              ;Br if successful
                                ;
                                ;  Error -- Unable to eliminate the region
                                ;
002030                                  FERR    #EM$UER         ;Unable to eliminate region
                                ;
                                ;  Finished
                                ;
002044  012602                  9$:     MOV     (SP)+,R2
002046  000207                          RETURN
```

```
                                        .SBTTL  INSTALL Command
                                ;-----------------------------------------------------------------------
                                ;  Process the INSTALL command.
                                ;
002050  004767  0000000      CMDINS: CALL    CVTTAB          ;Convert tabs to spaces
002054  004767  0000000              CALL    CVTUC           ;Convert to upper case
                                ;
                                ;  Accrue first keyword and see if it is ADD or DELETE
                                ;
002060  010302                  MOV     R3,R2           ;Save pointer to start of keyword
002062  004767  0000000          CALL    GTRD50          ;Accrue 1st keyword
002066  016700  0000000          MOV     R50BUF,R0       ;Get 1st 3 chars of keyword
002072  020067  175724           CMP     R0,R50ADD       ;Is keyword ADD?
002076  001421                  BEQ     INSADD          ;Br if yes
002100  020067  175720           CMP     R0,R50DEL       ;Is keyword DELETE?
002104  001473                  BEQ     INSDEL          ;Br if yes
002106  020067  175716           CMP     R0,R50REM       ;Is keyword REMOVE?
002112  001470                  BEQ     INSDEL          ;Br if yes
                                ;
                                ;  Keyword is not ADD or DELETE.
                                ;  Error if 1st keyword is longer than 2 characters.
                                ;
002114  010300                  MOV     R3,R0           ;Get pointer past end of keyword
002116  160200                  SUB     R2,R0           ;Determine length of 1st keyword
002120  020027  000002           CMP     R0,#2           ;1st keyword longer than 2 chars?
002124  101404                  BLOS    1$              ;Br if not
002126                          FABORT  #ILLCMD         ;Invalid command
                                ;
                                ;  This must be a device handler name -- Ignore this command
                                ;
002136  000167  0000000      1$:     JMP     RDCMD           ;Finished installing device handler
```

```
                                ;-------------------------------------------------------------------------
                                ;  We are adding an image
                                ;
002142  004767  000000G         INSADD: CALL    CKSYPV          ;Require SYSPRV privilege
002146  005067  175660                  CLR     IATTRB          ;Clear all attribute flags
002152  004767  000000G                 CALL    CLRPRV          ;Reset privilege flag cells
002156  004767  000546                  CALL    INSOPT          ;Process any options that follow ADD
                                ;
                                ;  Get the name of the image and see if it is currently in table
                                ;
002162  004767  000144                  CALL    INSNAM          ;Get name of image and look it up
002166  103002                          BCC     1$              ;Br if name already in table
                                ;
                                ;  Name is not currently in table.
                                ;  Try to find a free slot in the table.
                                ;
002170  004767  000450                  CALL    INSFRE          ;Find free table entry
                                ;
                                ;  Set up program name for free entry
                                ;
002174  012702  000000C         1$:     MOV     #II$NAM+IIBUF,R2;Point to target buffer
002200  012704  000000G                 MOV     #FILNAM,R4      ;Point to name
002204  012700  000004                  MOV     #4,R0           ;Move 4 words
002210  012422                  2$:     MOV     (R4)+,(R2)+     ;Move name to buffer
002212  077002                          SOB     R0,2$
                                ;
                                ;  At this point, R5 points to buffer where install entry is to be made
                                ;  Process any options that follow the file name
                                ;
002214  004767  000510                  CALL    INSOPT          ;Process any options that follow name
                                ;
                                ;  Move attribute and privilege flags into install entry
                                ;
002220  010546                          MOV     R5,-(SP)        ;Save address of entry in install table
002222  016767  175604  000000C         MOV     IATTRB,II$FLG+IIBUF ;Set attribute flags for program
002230  012702  0000000                 MOV     #PFS0,R2        ;Point to accrued privilege flags
002234  012703  000000C                 MOV     #II$PRV+IIBUF,R3;Install entry flags
002240  012704  0000000                 MOV     #PFC0,R4        ;Flags to clear when run
002244  012705  000000C                 MOV     #II$NPV+IIBUF,R5;Install entry
002250  012700  0000000                 MOV     #PVNPW,R0       ;Get # words to move
002254  012223                  3$:     MOV     (R2)+,(R3)+     ;Move flags to be set
002256  012425                          MOV     (R4)+,(R5)+     ;Move flags to be cleared
002260  077003                          SOB     R0,3$
002262  012605                          MOV     (SP)+,R5        ;Recover address of entry in install table
                                ;
                                ;  Add the entry to the install table
                                ;
002264  004767  000424                  CALL    INSPUT          ;Add entry to table
                                ;
                                ;  Finished
                                ;
002270  000167  0000000                 JMP     RDCMD
```

```
      1                                 ;
      2                                 ;  Process INSTALL DELETE command
      3                                 ;
      4 002274  004767  000000G         INSDEL: CALL    CKSYPV          ;Require SYSPRV privilege
      5                                 ;
      6                                 ;  See if we can find program in install table
      7                                 ;
      8 002300  004767  000026                  CALL    INSNAM          ;Accrue file spec and try to find in table
      9 002304  103004                          BCC     1$              ;Br if found file name in table
     10 002306                                  FABORT  #EM$FNI         ;File not installed
     11                                 ;
     12                                 ;  Found entry, delete it
     13                                 ;
     14 002316  005067  000000C         1$:     CLR     II$NAM+IIBUF    ;Say image is not installed
     15 002322  004767  000366                  CALL    INSPUT          ;Write entry back to table
     16                                 ;
     17                                 ;  Finished
     18                                 ;
     19 002326  000167  000000G                 JMP     RDCMD
```

```
                                ;*****************************************************************************
                                ;  Accrue a file spec for a file that is being installed and look the
                                ;  name up in the install table.
                                ;
                                ;  Inputs:
                                ;   R3 = Pointer to start of file spec.
                                ;
                                ;  Outputs:
                                ;   C-flag cleared ==> Found entry in install table.
                                ;   C-flag set ==> Cannot find entry in install table.
                                ;   R5 = Address of entry in install table if found.
                                ;
002332  010246                  INSNAM: MOV     R2,-(SP)
002334  010446                          MOV     R4,-(SP)
                                ;
                                ;  Determine if wildcard ("*") specified as device name
                                ;
002336  005002                          CLR     R2              ;Assume device name is not wild
002340  004767  0000000                 CALL    SKPSPC          ;Skip up to start of file spec
002344  121327  000052                  CMPB    (R3),#'*        ;Wildcard as device name?
002350  001013                          BNE     11$             ;Br if not
002352  126327  000001  000072          CMPB    1(R3),#':       ;Is this the device name?
002360  001007                          BNE     11$             ;Br if not
002362  012702  132500                  MOV     #WLDNAM,R2      ;Remember device name is wild
002366  112713  000040                  MOVB    #' ,(R3)        ;Blank out the device name
002372  112763  000040  000001          MOVB    #' ,1(R3)
                                ;
                                ;  Accrue the file spec
                                ;
002400  012704  0000000         11$:    MOV     #R50SAV,R4      ;Get default file extension (SAV)
002404  005005                          CLR     R5              ;Say this is an input file
002406  004767  0000000                 CALL    ACRFIL          ;Accrue the file spec
002412  103510                          BCS     10$             ;Br if invalid file spec
002414  005767  0000020                 TST     FILNAM+2        ;Was a file name specified?
002420  001505                          BEQ     10$             ;Br if not
002422  005702                          TST     R2              ;Is device name wild?
002424  001403                          BEQ     12$             ;Br if not
002426  010267  0000000                 MOV     R2,FILNAM       ;Set wildcard as file name
002432  000431                          BR      5$
                                ;
                                ;  Translate logical file device to physical device with unit #
                                ;
002434  012705  0000000         12$:    MOV     #FILNAM,R5      ;Point to file spec
002440  004767  0000000                 CALL    LOGASN          ;Do logical assignment
002444  010346                          MOV     R3,-(SP)        ;SAVE POINTER
002446  010503                          MOV     R5,R3           ;GET COPY OF DEVICE POINTER
002450  004767  0000000                 CALL    FORCE0          ;MAKE BLANK UNIT = 0
002454  012603                          MOV     (SP)+,R3        ;RESTORE POINTER
                                ;
                                ;  Error if logical disk (LD) specified for device
                                ;
002456  016705  0000000                 MOV     FILNAM,R5       ;Get the device name
002462  004767  0000000                 CALL    CHKDEV          ;Convert to device index number
002466  103407                          BCS     6$              ;Br if invalid device
002470  020467  0000000                 CMP     R4,LDDEVX       ;Is this a logical disk?
002474  001010                          BNE     5$              ;Br if not
002476                                  FABORT  #EM$LDI         ;Can't have installed program on LD
```

```
     58 002506                                  6$:     FABORT  #EM$IAD         ;Invalid device
     59                                         ;
     60                                         ;  Try to find existing entry for file in install table
     61                                         ;
     62 002516  016705  0000000G                5$:     MOV     INSTBL,R5       ;Point to 1st entry in table
     63 002522  010567  0000000G                1$:     MOV     R5,INGADR       ;Set address in EMT
     64 002526  012700  0000000G                        MOV     #INGEMT,R0      ;Point to get EMT
     65 002532  104375                                  EMT     375             ;Get next install table entry
     66 002534  005760  000000C                         TST     IIBUF+II$NAM(R0);Is this entry empty?
     67 002540  001422                                  BEQ     2$              ;Br if yes
     68 002542  012700  000006                          MOV     #6,R0           ;Get index to last word of name
     69 002546  026027  0000000G 132500   3$:     CMP     FILNAM(R0),#WLDNAM ;Wildcard?
     70 002554  001410                                  BEQ     13$             ;Br if yes
     71 002556  026027  000000C 132500                 CMP     IIBUF+II$NAM(R0),#WLDNAM ;Wildcard?
     72 002564  001404                                  BEQ     13$             ;Br if yes
     73 002566  026060  0000000G 000000C               CMP     FILNAM(R0),IIBUF+II$NAM(R0) ;Compare names
     74 002574  001004                                  BNE     2$              ;Br if this is not right entry
     75 002576  162700  000002             13$:     SUB     #2,R0           ;More words to check?
     76 002602  002361                                  BGE     3$              ;Br if yes
     77 002604  000407                                  BR      4$              ;Found entry
     78 002606  062705  0000000G               2$:     ADD     #II$$SZ,R5      ;Point to next entry in table
     79 002612  020567  0000000G                        CMP     R5,INSTBN       ;Reached end of table?
     80 002616  103741                                  BLO     1$              ;Loop if not
     81                                         ;
     82                                         ;  Cannot find entry in table
     83                                         ;
     84 002620  000261                                  SEC                     ;Signal failure on return
     85 002622  000401                                  BR      9$
     86                                         ;
     87                                         ;  We found entry for this name
     88                                         ;
     89 002624  000241                          4$:     CLC                     ;Signal success on return
     90                                         ;
     91                                         ;  Finished
     92                                         ;
     93 002626  012604                          9$:     MOV     (SP)+,R4
     94 002630  012602                                  MOV     (SP)+,R2
     95 002632  000207                                  RETURN
     96                                         ;
     97                                         ;  Invalid file spec
     98                                         ;
     99 002634                                  10$:    FABORT  #BDFNAM         ;Invalid file spec
```

```
                                ;-----------------------------------------------------------------------------
                                ;  Try to find a free entry in the install table.
                                ;
                                ;  Outputs:
                                ;   R5 = Address of free entry in install table.
                                ;
002644  016705  0000000G      INSFRE: MOV     INSTBL,R5       ;Point to 1st install table entry
002650  010567  0000000G      1$:     MOV     R5,INGADR       ;Set address for EMT
002654  012700  0000000G              MOV     #INGEMT,R0      ;EMT to get assign table entry
002660  104375                      EMT     375             ;Get next entry
002662  005767  000000C             TST     II$NAM+IIBUF    ;Is this entry free?
002666  001411                      BEQ     2$              ;Br if found free entry
002670  062705  0000000G            ADD     #II$$SZ,R5      ;Point to next entry
002674  020567  0000000G            CMP     R5,INSTBN       ;Reached end of table?
002700  103763                      BLO     1$              ;Loop if not
                                ;
                                ;  No free table entries
                                ;
002702                                  FABORT  #EM$ITF         ;Install table full
                                ;
                                ;  Found free entry
                                ;
002712  000207                 2$:     RETURN
```

```
      1                                 ;-----------------------------------------------------------------------
      2                                 ;  Write the current install entry into the install table.
      3                                 ;
      4                                 ;  Inputs:
      5                                 ;   R5 = Address in install table where entry is to go.
      6                                 ;
      7 002714  010567  0000000         INSPUT: MOV     R5,INPADR       ;Set address for EMT
      8 002720  012700  0000000                 MOV     #INPEMT,R0      ;Point to EMT
      9 002724  104375                          EMT     375             ;Store install table entry
     10 002726  000207                          RETURN
```

```
                                ;-----------------------------------------------------------------------
                                ;  Process command options specified with an INSTALL command.
                                ;
                                ;  Inputs:
                                ;   R3 = Pointer to option list.
                                ;
                                ;  Outputs:
                                ;   R3 = Points past end of option list
                                ;
002730  010446                  INSOPT: MOV     R4,-(SP)
                                ;
                                ;  Process list of options
                                ;
002732  012704  000000'                 MOV     #INSHD,R4       ;Point to table of options
002736  004767  0000000                 CALL    OPTLST          ;Process an option list
                                ;
                                ;  Finished
                                ;
002742  012604                          MOV     (SP)+,R4
002744  000207                          RETURN

                                ;-----------------------------------------------------------------------
                                ;  Subroutine called to set a specific install attribute flag
                                ;
                                ;  Inputs:
                                ;   R4 = Pointer to option table entry.
                                ;
                                ;  Outputs:
                                ;    IATTRB = Combined attribute flags
                                ;
002746  051467  175060          INSATR: BIS     (R4),IATTRB     ;Set attribute flag
002752  000207                          RETURN                  ;Finished
```

```
                                        .SBTTL  MOUNT command
                                ;-----------------------------------------------------------------------
                                ;  Mount a file structure.
                                ;  This command has two possible forms:
                                ;    MOUNT dev -- Simply enable directory caching
                                ;    MOUNT[/[NO]WRITE] LDn file-spec [logical-name] -- Mount logical disk
                                ;
002754  004767  000000G       CMDMNT: CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACES
                                ;
                                ;  Determine if a command switch was specified
                                ;
002760  105067  000000G               CLRB    NOFLAG          ;ASSUME /NOWRITE NOT WANTED
002764  122327  000040        1$:     CMPB    (R3)+,#' '      ;SKIP ANY LEADING SPACES
002770  001775                        BEQ     1$
002772  124327  000057                CMPB    -(R3),#'/       ;WAS A SWITCH SPECIFIED?
002776  001014                        BNE     3$              ;BR IF NOT
003000  005203                        INC     R3              ;SKIP PAST /
003002  012704  000160'              MOV     #MTOPHD,R4      ;POINT TO TABLE OF SWITCHES
003006  004767  000000G               CALL    SEARCH          ;LOOK UP THE SWITCH
003012  103004                        BCC     2$              ;BR IF FOUND
003014                                FABORT  #INVOPT         ;INVALID SWITCH
003024  111467  000000G       2$:     MOVB    (R4),NOFLAG     ;REMEMBER IF NOWRITE SEEN
                                ;
                                ;  Get device name
                                ;
003030  105713                3$:     TSTB    (R3)            ;IS THERE A DEVICE NAME?
003032  001004                        BNE     12$             ;BR IF YES
003034                                FABORT  #ILLCMD         ;INVALID COMMAND
003044  004767  000000G       12$:    CALL    GTRD50          ;GET DEVICE NAME
003050  016702  000000G               MOV     R50BUF,R2       ;PICK UP DEVICE NAME
003054  020267  000000G               CMP     R2,R50LD        ;IS NAME "LD"?
003060  001002                        BNE     15$             ;BR IF NOT
003062  016702  000000G               MOV     R50LD0,R2       ;TRANSLATE LD TO LD0
003066  010267  000000G       15$:    MOV     R2,MNTDEV       ;SAVE DEVICE NAME
003072  122327  000072                CMPB    (R3)+,#':       ;WAS DEVICE NAME TERMINATED WITH COLON?
003076  001401                        BEQ     4$              ;BR IF YES
003100  005303                        DEC     R3              ;BACKUP POINTER IF NOT
                                ;
                                ;  See if device we are mounting is allocated to another user
                                ;
003102  016767  000000G 000000G 4$:     MOV     MNTDEV,ALCDEV   ;Set device name for allocation EMT
003110  012700  000000G               MOV     #TALEMT,R0      ;Point to EMT argument block
003114  104375                        EMT     375             ;Check for allocation conflict
003116  103022                        BCC     5$              ;Br if no allocation failure
003120  123727  000000G 000001        CMPB    @#ERRLOC,#1     ;Device allocated to another user?
003126  001016                        BNE     5$              ;Br if not
003130  010005                        MOV     R0,R5           ;Save # of job that owns the device
003132                                FERR    #EM$DAA         ;Device allocated to job
003146  004767  000000G               CALL    PRTDEC          ;Print # of job that owns device
003152                                .PRINT  #CRLF           ;End print line
003160  000167  000000G               JMP     RDCMD           ;Abort command
                                ;
                                ;  See if there is a file name specified.
                                ;
003164  004767  000000G       5$:     CALL    SKPSPC          ;Skip any spaces
003170  105713                        TSTB    (R3)            ;Was a file name specified?
003172  001002                        BNE     MNTLD           ;BR IF YES
```

```
     58 003174  000167  000346                    JMP     MNTCSH          ;NO FILE NAME -- GO ENABLE DIRECTORY CACHING
```

```
                                        ;
                                        ;  This is a MOUNT command with a file name.
                                        ;  We are mounting a logical disk.
                                        ;  Make sure Logical Disk support was genned into system.
                                        ;
003200  105767  000000G          MNTLD:  TSTB    VLDSYS          ;WAS LD SUPPORT GENNED IN?
003204  001004                          BNE     18$             ;BR IF YES
003206                                  FABORT  #EM$NLD         ;LD SUPPORT NOT GENNED IN
                                        ;
                                        ;  Make sure the logical disk name is ok.
                                        ;
003216  010200                  18$:    MOV     R2,R0           ;GET LD DEVICE NAME
003220  004767  000412                  CALL    LDDEVN          ;GET LD DEVICE INDEX NUMBER
003224  103004                          BCC     6$              ;BR IF OK
003226                                  FABORT  #INVLDN         ;ERROR IF NAME IS NOT LD
003236  010002                  6$:     MOV     R0,R2           ;CARRY INDEX IN R2
                                        ;
                                        ;  Close the log file if it is on the logical disk
                                        ;
003240  016705  0000000                 MOV     MNTDEV,R5       ;GET NAME OF LOGICAL DEVICE
003244  004767  0000000                 CALL    LOGCHK          ;SEE IF WE NEED TO CLOSE THE LOG FILE
                                        ;
                                        ;  Tell system to stop doing directory caching for the device
                                        ;
003250  112767  000001  0000000         MOVB    #1,SERFLG       ;DO .SERR TO AVOID ABORT FOR ILLEGAL DEVICE
003256  012700  0000000                 MOV     #DMTARG,R0      ;POINT TO EMT ARGUMENT BLOCK
003262  104375                          EMT     375             ;TELL SYSTEM TO STOP DOING CACHING
003264  105067  0000000                 CLRB    SERFLG          ;DO .HERR
                                        ;
                                        ;  Accrue the file name
                                        ;
003270  012704  0000000                 MOV     #R50DSK,R4      ;SET DEFAULT FILE EXTENSION
003274  005005                          CLR     R5              ;SAY THIS IS AN INPUT FILE
003276  004767  0000000                 CALL    ACRFIL          ;ACCRUE THE FILE SPEC
003302  103004                          BCC     13$             ;BR IF OK
003304                          5$:     FABORT  #BDFNAM         ;INVALID FILE SPEC
                                        ;
                                        ;  Translate logical file device name to physical device
                                        ;
003314  012705  0000000         13$:    MOV     #FILNAM,R5      ;POINT TO FILE SPEC AREA
003320  004767  0000000                 CALL    LOGASN          ;PERFORM ANY LOGICAL DEVICE ASSIGNMENT
                                        ;
                                        ;  Make sure file is not on same LD as is being mounted
                                        ;
003324  016700  0000000                 MOV     FILNAM,R0       ;GET DEVICE THAT FILE IS ON
003330  004767  000302                  CALL    LDDEVN          ;SEE IF FILE IS ON A LD DEVICE
003334  103406                          BCS     17$             ;BR IF NOT ON LD DEVICE
003336  020002                          CMP     R0,R2           ;IS IT ON LOWER # LD?
003340  103404                          BLO     17$             ;BR IF YES
003342                                  FABORT  #INVLDM         ;INVALID LD ORDER
                                        ;
                                        ;  Make sure the file name is not null
                                        ;
003352  005767  0000020         17$:    TST     FILNAM+2        ;Is file name null?
003356  001752                          BEQ     5$              ;Br if yes -- Error
                                        ;
                                        ;  Move file name into logical disk name table
```

```
                                ;
003360  010205                          MOV     R2,R5           ;GET LOGICAL DISK UNIT #
003362  006305                          ASL     R5              ;*4 WORDS PER ENTRY
003364  006305                          ASL     R5
003366  062705  000000G                 ADD     #LDNAME,R5      ;POINT TO ENTRY IN LOGICAL DISK TABLE
003372  012704  000000G                 MOV     #FILNAM,R4      ;POINT TO FILE NAME BUFFER
003376  012425                          MOV     (R4)+,(R5)+     ;MOVE NAME TO TABLE
003400  012425                          MOV     (R4)+,(R5)+
003402  012425                          MOV     (R4)+,(R5)+
003404  011415                          MOV     (R4),(R5)
                                ;
                                ;  Set up flags for logical disk
                                ;
003406  005004                          CLR     R4
003410  105767  000000G                 TSTB    NOFLAG          ;WAS NOWRITE SPECIFIED?
003414  001402                          BEQ     9$              ;BR IF NOT
003416  052704  000000G                 BIS     #LD$RON,R4      ;SET READ-ONLY FLAG
003422  010462  000000G         9$:     MOV     R4,LDFLAG(R2)
                                ;
                                ;  Lookup file and set up information about position of logical
                                ;  disk on physical disk.
                                ;
003426  004767  000000G                 CALL    LDMNT           ;SET UP INFO ABOUT FILE FOR LOGICAL DISK
003432  103004                          BCC     10$             ;BR IF FOUND FILE
003434                                  FABORT  #CSIMS4         ;CAN'T FIND FILE
                                ;
                                ;  Protect the file
                                ;
003444  010205                  10$:    MOV     R2,R5           ;GET LOGICAL DISK UNIT #
003446  006305                          ASL     R5              ;* 4 WORDS PER ENTRY
003450  006305                          ASL     R5
003452  062705  000000G                 ADD     #LDNAME,R5      ;POINT TO ENTRY WITH FILE NAME
003456  112767  000001  000000G         MOVB    #1,SERFLG       ;DO .SERR TO AVOID ABORTS ON ERRORS
003464                                  .FPROT  #XAREA,#1,R5,#1
003510  105067  000000G                 CLRB    SERFLG          ;DO .HERR
                                ;
                                ;  See if a logical device name was specified following the file name
                                ;
003514  122327  000040          16$:    CMPB    (R3)+,#' '      ;SKIP BLANKS
003520  001775                          BEQ     16$
003522  105743                          TSTB    -(R3)           ;WAS A LOGICAL DEVICE NAME SPECIFIED?
003524  001410                          BEQ     MNTCSH          ;BR IF NOT
                                ;
                                ;  Assign logical name to logical disk
                                ;
003526  004767  000000G                 CALL    GTRD50          ;ACCRUE LOGICAL DEVICE NAME
003532  016705  000000G                 MOV     R50BUF,R5       ;GET LOGICAL DEVICE NAME
003536  016700  000000G                 MOV     MNTDEV,R0       ;GET PHYSICAL DEVICE NAME
003542  004767  000000G                 CALL    DOASGN          ;DO THE ASSIGNMENT
                                ;
                                ;  Initiate directory caching for the device whose name is in MNTDEV
                                ;
003546  012700  000000G         MNTCSH: MOV     #MNTARG,R0      ;POINT TO MOUNT ARGUMENT BLOCK
003552  004767  000010                  CALL    DOMNT           ;ENABLE CASHING FOR DEVICE
                                ;
                                ;  Now do a SET LD CLEAN operation to make sure all logical disks
                                ;  are set up correctly
```

```
    115                                 ;
    116 003556  004767  0000000                 CALL    LDCLEN          ;DO SET LD CLEAN
    117                                 ;
    118                                 ;  Finished with MOUNT command
    119                                 ;
    120 003562  000167  0000000                 JMP     RDCMD
```

```
                                ;-----------------------------------------------------------------------
                                ;  DOMNT is called to execute a mount or ,dismount EMT and check for
                                ;  errors.  If an error is detected, an error message is generated.
                                ;
                                ;  Inputs:
                                ;   R0 = Address of EMT argument block for mount or dismount emt.
                                ;
003566  104375                  DOMNT:  EMT     375             ;DO THE EMT (MOUNT OR DISMOUNT)
003570  103021                          BCC     9$              ;BR IF NO ERROR
                                ;
                                ;  Error on mount or dismount emt.
                                ;
003572  123727  0000000 000001          CMPB    @#ERRLOC,#1     ;MOUNT TABLE OVERFLOW?
003600  001007                          BNE     2$              ;BR IF NOT
003602                                  FWARN   #MNTFUL         ;MOUNT TABLE OVERFLOW
003616  000406                          BR      9$
003620                          2$:     FERR    #BDFNAM         ;INVALID DEVICE NAME
                                ;
                                ;  Finished
                                ;
003634  000207                  9$:     RETURN

                                ;-----------------------------------------------------------------------
                                ;  LDDEVN is called to determine if a device name is of the form LDn and
                                ;  if so, to determine the unit number.
                                ;
                                ;  Inputs:
                                ;   R0 = Device name (rad50)
                                ;
                                ;  Outputs:
                                ;   C-flag cleared ==> Device is LDn
                                ;   C-flag set     ==> Device is not LDn
                                ;   R0 = LD device index number (2*n)
                                ;
003636  020067  0000000         LDDEVN: CMP     R0,R50LD        ;Is name "LD"?
003642  001002                          BNE     1$              ;Br if not
003644  016700  0000000                 MOV     R50LD0,R0       ;Replace "LD" by "LD0"
003650  020067  0000000         1$:     CMP     R0,R50LD0       ;Is name in the range LD0 to LD7?
003654  103410                          BLO     2$              ;Br if not
003656  020067  0000000                 CMP     R0,R50LD7
003662  101005                          BHI     2$
003664  166700  0000000                 SUB     R50LD0,R0       ;Get unit number only
003670  006300                          ASL     R0              ;Convert to device index number
003672  000241                          CLC                     ;Signal success on return
003674  000401                          BR      3$
003676  000261                  2$:     SEC                     ;Signal failure on return
003700  000207                  3$:     RETURN
```

```
                                .SBTTL  DISMOUNT command
                        ;-----------------------------------------------------------------------
                        ;  Dismount a file structure.
                        ;
003702  004767  0000000 CMDDMT: CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACES
                        ;
                        ;  See if any qualifier were specified following command keyword
                        ;
003706  105067  174122          CLRB    DMTNLG          ;Clear flag set if /NOLOG specified
003712  004767  0000000        CALL    SKPSPC          ;Skip over any spaces
003716  121327  000057         CMPB    (R3),#'/        ;Any qualifiers here?
003722  001004                 BNE     5$              ;Br if not
                        ;
                        ;  Process qualifier following command keyword
                        ;
003724  012704  000174'        MOV     #DMTQHD,R4      ;Point to parsing table
003730  004767  0000000        CALL    OPTLST          ;Process any qualifiers
                        ;
                        ;  Skip leading spaces and make sure a device name was specified
                        ;
003734  004767  0000000 5$:     CALL    SKPSPC          ;SKIP LEADING SPACES
003740  105713                 TSTB    (R3)            ;WAS A DEVICE NAME SPECIFIED
003742  001004                 BNE     6$              ;BR IF YES
003744                  7$:     FABORT  #ILLCMD         ;INVALID COMMAND IF NO DEVICE NAME
                        ;
                        ;  Get name of device that is being dismounted
                        ;
003754  004767  0000000 6$:     CALL    GTRD50          ;ACCRUE THE DEVICE NAME
003760  016700  0000000        MOV     R50BUF,R0       ;GET DEVICE NAME
003764  001767                 BEQ     7$              ;BR IF NO DEVICE NAME
003766  004767  0000000 2$:     CALL    ASNSRC          ;SEE IF IT IS A LOGICAL NAME
003772  103403                 BCS     1$              ;BR IF NOT
003774  016200  0000000        MOV     AT$DEV(R2),R0   ;GET ASSIGNED NAME
004000  000772                 BR      2$              ;ALLOW INDIRECT ASSIGNS
004002  020067  0000000 1$:     CMP     R0,R50LD        ;IS NAME "LD"?
004006  001002                 BNE     4$              ;BR IF NOT
004010  016700  0000000        MOV     R50LD0,R0       ;TRANSLATE LD TO LD0
004014  010067  0000000 4$:     MOV     R0,MNTDEV       ;SAVE NAME OF DEVICE BEING DISMOUNTED
                        ;
                        ;  Close the log file if it is on the device being dismounted
                        ;
004020  016705  0000000        MOV     MNTDEV,R5       ;GET NAME OF DEVICE BEING DISMOUNTED
004024  004767  0000000        CALL    LOGCHK          ;SEE IF LOG FILE IS ON THAT DEVICE
                        ;
                        ;  Print information message if device is still mounted by other jobs
                        ;
004030  105767  174000         TSTB    DMTNLG          ;Was /NOLOG qualifier specified?
004034  001013                 BNE     3$              ;Br if yes -- Suppress information message
004036  016705  0000000        MOV     MNTDEV,R5       ;GET NAME OF DEVICE BEING DISMOUNTED
004042  004767  0000000        CALL    CHKMNT          ;IS THIS DEVICE MOUNTED?
004046  103406                 BCS     3$              ;BR IF NOT
004050  004767  0000000        CALL    CHKMTX          ;IS DEVICE MOUNTED BY ANYONE OTHER THAN US?
004054  103003                 BCC     3$              ;BR IF NOT
004056                          .PRINT  #INFOMT         ;STILL MOUNTED BY OTHER USERS
                        ;
                        ;  Tell system to stop doing directory caching for the device
                        ;
```

```
     58 004064  112767  000001  000000G 3$:     MOVB    #1,SERFLG       ;DO .SERR TO AVOID ABORT FOR ILLEGAL DEVICE
     59 004072  012700  000000G                 MOV     #DMTARG,R0      ;POINT TO EMT ARGUMENT BLOCK
     60 004076  104375                          EMT     375             ;TELL SYSTEM TO STOP DOING CACHING
     61 004100  105067  000000G                 CLRB    SERFLG          ;DO .HERR
     62                                 ;
     63                                 ;  See if a logical disk is being dismounted
     64                                 ;
     65 004104  016702  000000G                 MOV     MNTDEV,R2       ;GET DEVICE NAME
     66 004110  020267  000000G                 CMP     R2,R50LD0       ;IS THIS A LOGICAL DISK?
     67 004114  103420                          BLO     9$              ;BR IF NOT
     68 004116  020267  000000G                 CMP     R2,R50LD7
     69 004122  101015                          BHI     9$
     70 004124  166702  000000G                 SUB     R50LD0,R2       ;GET UNIT #
     71 004130  006302                          ASL     R2              ;CONVERT TO WORD INDEX
     72 004132  005062  000000G                 CLR     LDPDEV(R2)      ;NO PHYSICAL DEVICE ASSIGNMENT
     73 004136  006302                          ASL     R2              ;GET INDEX INTO LDNAME TABLE
     74 004140  006302                          ASL     R2
     75 004142  005062  000000G                 CLR     LDNAME(R2)      ;NO FILE NAME
     76 004146  016700  000000G                 MOV     MNTDEV,R0       ;GET NAME OF LOGICAL DISK
     77 004152  004767  000000G                 CALL    DEADEV          ;DEASSIGN ANY LOGICAL NAMES
     78                                 ;
     79                                 ;  Finished
     80                                 ;
     81 004156  004767  000000G         9$:     CALL    LDCLEN          ;RESET INFORMATION ABOUT LOGICAL DISKS
     82 004162  000167  000000G                 JMP     RDCMD           ;FINISHED WITH COMMAND
     83                                 ;
     84                                 ;  Process /NOLOG qualifier for DISMOUNT command
     85                                 ;
     86 004166  105267  173642          DMTNLQ: INCB    DMTNLG          ;Remember qualifier specified
     87 004172  000207                  DMTLQQ: RETURN
```

```
                        .SBTTL  INITIALIZE command
                        .SBTTL  SQUEEZE command
                        .SBTTL  FORMAT command
                ;*****************************************************************************
                ;  Disallow INITIALIZE and SQUEEZE commands if any other users have
                ;  the specified device mounted.
                ;
004174          CMDINI:
004174          CMDSQZ:
004174          CMDFMT:
                ;
                ;  See if user is authorized to use these commands
                ;
004174  032767  000000C 000000G         BIT     #PO$NFW!PO$BYP,PRIVCO   ;Authorized for these commands?
004202  001004                          BNE     13$             ;Br if yes
004204                                  FABORT  #EM$NFW         ;Not authorized for this command
                ;
                ;  Skip over any options specified with command keyword
                ;
004214  004767  0000000         13$:    CALL    CVTTAB          ;Convert tabs to spaces
004220  121327  000057                  CMPB    (R3),#'/        ;Were any keyword options specified?
004224  001005                          BNE     10$             ;Br if not
004226  112300                  1$:     MOVB    (R3)+,R0        ;Get next character from command
004230  001407                          BEQ     15$             ;Br if no device was specified
004232  120027  000040                  CMPB    R0,#40          ;Is this a space?
004236  001373                          BNE     1$              ;Keep looking if not
                ;
                ;  Accrue the device name
                ;
004240  004767  0000000         10$:    CALL    SKPSPC          ;Skip over any spaces
004244  105713                          TSTB    (R3)            ;Was a device name specified?
004246  001004                          BNE     14$             ;Br if yes
004250                          15$:    FABORT  #EM$DNR         ;Device name is required
004260  004767  0000000         14$:    CALL    GTRD50          ;Accrue the device name
004264  016705  0000000                 MOV     R50BUF,R5
004270  010567  0000000                 MOV     R5,MNTDEV       ;Set name for mount/dismount
004274  010567  173500                  MOV     R5,SQZDEV       ;Save name of device being squeezed
004300  010567  0000000                 MOV     R5,ALCDEV       ;Set name for test-allocation EMT
                ;
                ;  Disallow SQUEEZE/INITIALIZE/FORMAT of any disk with a system file
                ;
004304  004767  000232                  CALL    CKSYDV          ;See if device has any system files
004310  103004                          BCC     11$             ;Br if not
004312                                  FABORT  #EM$SSY         ;Can't do this to SY
                ;
                ;  See if this device is mounted by anyone
                ;
004322  004767  0000000         11$:    CALL    CHKMNT          ;See if device is mounted by anyone
004326  103412                          BCS     8$              ;Br if not
                ;
                ;  This device is in the mount table (R5=pointer to entry).
                ;  See if this device is mounted by any other users
                ;
004330  004767  0000000                 CALL    CHKMTX          ;Is device mounted by any other users?
004334  103004                          BCC     2$              ;Br if not
004336                          3$:     FABORT  #OTRMNT         ;Command not legal if mounted by others
                ;
```

```
                                        ;  Dismount and remount this device to clean out all file entries
                                        ;  associated with this device.
                                        ;
004346  012700  0000000G      2$:       MOV     #MNTARG,R0      ;Remount the device
004352  104375                          EMT     375
                                        ;
                                        ;  See if this device is allocated to any other user
                                        ;
004354  120467  0000000G      8$:       CMPB    R4,LDDEVX       ;Is this a logical disk?
004360  001404                          BEQ     12$             ;Br if yes -- Don't check for allocation
004362  016700  0000000G              MOV     ALCDEV,R0       ;Get RAD50 device name
004366  004767  0000000G              CALL    CHKALC          ;See if device is allocated to anyone else
                                        ;
                                        ;  Remove from mount table any logical disks that are on the device
                                        ;  being squeezed or initialized.
                                        ;
004372  016705  173402      12$:      MOV     SQZDEV,R5       ;Get back device name
004376  004767  0000000G              CALL    CHKDEV          ;Convert device name to dev # & unit #
004402  020467  0000000G              CMP     R4,LDDEVX       ;Is this a logical disk?
004406  001406                          BEQ     4$              ;Br if yes
004410  005002                          CLR     R2              ;Say base block # = 0
004412  012703  177777                  MOV     #177777,R3      ;Say top block = 177777
004416  000300                          SWAB    R0              ;Put unit # in high byte
004420  050004                          BIS     R0,R4           ;Get unit # and device # together
004422  000410                          BR      5$
004424  006300              4$:       ASL     R0              ;Convert unit # to word table index
004426  016004  0000000G              MOV     LDPDEV(R0),R4   ;Get physical dev # and unit #
004432  016002  0000000G              MOV     LDBASE(R0),R2   ;Get base block of logical disk
004436  010203                          MOV     R2,R3
004440  066003  0000000G              ADD     LDSIZE(R0),R3   ;Get top block number of logical disk
                                        ;
                                        ;  Search for any mounted logical disks that are on the disk being squeezed
                                        ;
004444  016705  0000000G      5$:       MOV     CSHDEV,R5       ;Point to start of mount table
004450  010500              7$:       MOV     R5,R0           ;Get address of mount entry
004452  004767  0000000G              CALL    CDGET           ;Copy mount entry into CDBUF
004456  020467  000000C               CMP     R4,CDBUF+CD$DVU ;Is this device on same physical device?
004462  001014                          BNE     6$              ;Br if not
004464  016700  000000C               MOV     CDBUF+CD$BAS,R0 ;Is this a logical disk?
004470  001411                          BEQ     6$              ;Br if not
004472  020002                          CMP     R0,R2           ;See if logical disk is within disk being sqze
004474  101407                          BLOS    6$              ;Br if not
004476  020003                          CMP     R0,R3           ;Compare upper range
004500  103005                          BHIS    6$
                                        ;
                                        ;  We found a logical disk within the disk being squeezed.
                                        ;  See if anyone else has that logical disk mounted.
                                        ;
004502  004767  0000000G              CALL    CHKMTX          ;Anyone else have this logical disk mounted?
004506  103713                          BCS     3$              ;Error if yes
                                        ;
                                        ;  Dismount this logical disk and any files associated with it
                                        ;
004510  004767  0000000G              CALL    DMTSUB          ;Do the dismount
                                        ;
                                        ;  Check for more nested logical disks
                                        ;
```

```
    115 004514  062705  0000000         6$:     ADD     #CD$$SZ,R5      ;Point to next mount table entry
    116 004520  020567  0000000                 CMP     R5,CSHDVN       ;Reached end of table?
    117 004524  103751                          BLO     7$              ;Loop if not
    118                                 ;
    119                                 ;  See if log file is opened to device being squeezed
    120                                 ;  If so, close it.
    121                                 ;
    122 004526  016705  173246                  MOV     SQZDEV,R5       ;Get name of device being squeezed
    123 004532  004767  0000000                 CALL    LOGCHK          ;See if log file is on that device
    124                                 ;
    125                                 ;  Call CCL to process the command
    126                                 ;
    127 004536  000167  0000000         9$:     JMP     CMDCCL          ;Call CCL to process the command
```

```
                                ;-------------------------------------------------------------------------------
                                ;  Check to see if a device being squeezed or initialized contains any
                                ;  system files.
                                ;
                                ;  Inputs:
                                ;   R5 = Rad50 device name of device being squeezed or initialized
                                ;
                                ;  Outputs:
                                ;   C-flag set ==> This device contains system files
                                ;
004542  010246                  CKSYDV: MOV     R2,-(SP)
004544  010346                          MOV     R3,-(SP)
004546  010446                          MOV     R4,-(SP)
004550  010546                          MOV     R5,-(SP)
                                ;
                                ;  Convert device name to device index and unit number
                                ;
004552  004767  0000000                 CALL    CHKDEV          ;Convert name to device index & unit #
004556  103422                          BCS     8$              ;Br if device name not recognized
                                ;
                                ;  At this point: R0 = unit number, R4 = device index number.
                                ;  Begin loop to check each system file savestatus block.
                                ;
004560  012705  004640'                 MOV     #SYFLVC,R5      ;Point to vector of savestatus addresses
004564  012503                  1$:     MOV     (R5)+,R3        ;Point to next savestatus block
004566  001416                          BEQ     8$              ;Br if reached end of list
004570  016302  0000000                 MOV     C.CSW(R3),R2    ;Get 1st word of channel
004574  042702  177701                  BIC     #^C<76>,R2      ;Extract device index number
004600  020204                          CMP     R2,R4           ;Does it match device being squeezed?
004602  001370                          BNE     1$              ;Br if not
004604  116302  0000000                 MOVB    C.DEVQ(R3),R2   ;Get unit number
004610  042702  177770                  BIC     #^C<7>,R2       ;Extract unit number
004614  020200                          CMP     R2,R0           ;Same as unit of device being squeezed?
004616  001362                          BNE     1$              ;Br if not
                                ;
                                ;  This device contains a system file
                                ;
004620  000261                          SEC                     ;Signal error on return
004622  000401                          BR      9$
                                ;
                                ;  This device does not contain a system file
                                ;
004624  000241                  8$:     CLC                     ;Signal no error
                                ;
                                ;  Finished
                                ;
004626  012605                  9$:     MOV     (SP)+,R5
004630  012604                          MOV     (SP)+,R4
004632  012603                          MOV     (SP)+,R3
004634  012602                          MOV     (SP)+,R2
004636  000207                          RETURN
                                ;
                                ;  List of addresses of savestatus blocks for system files
                                ;
004640  0000000                 SYFLVC: .WORD   SWPCHN           ;Swap file
004642  0000000                         .WORD   SEGCHN          ;PLAS swap file
004644  0000000                         .WORD   SPLCHN          ;Spool file
```

```
     58 004646  000000                    .WORD   KMNCHN          ;TSKMON.SAV
     59 004650  000000                    .WORD   CCLSAV          ;CCL.SAV
     60 004652  000000                    .WORD   INDSAV          ;IND.SAV
     61 004654  000000                    .WORD   INDTSV          ;TSXIND.TSX
     62 004656  000000                    .WORD   0               ;End of list
```

```
                                        .SBTTL  MONITOR command
                                ;-------------------------------------------------------------------------------
                                ;  The MONITOR command is used to initiate performance monitoring.
                                ;  The form of the command is:
                                ;  MONITOR base-address,top-address[,#-bytes-per-cell]/switches
                                ;
004660  005067  000000G        CMDMON: CLR     MNFLGS          ;INITIALIZE EMT ARG BLOCK
004664  005067  000000G                CLR     MNBPC
004670  004767  000000G                CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACES
004674  004767  000000G                CALL    DELSPC          ;DELETE SPACES FROM COMMAND LINE
004700  004767  000000G                CALL    ACROCT          ;ACCRUE BASE ADDRESS
004704  010167  000000G                MOV     R1,MNBASE       ;SET IN EMT ARG BLOCK
004710  122327  000054                CMPB    (R3)+,#',       ;COMMA SHOULD BE DELIMITER
004714  001075                        BNE     3$
004716  004767  000000G                CALL    ACROCT          ;ACCRUE TOP ADDRESS
004722  010167  000000G                MOV     R1,MNTOP
004726  112300                        MOVB    (R3)+,R0        ;GET DELIMITER
004730  001430                        BEQ     1$              ;BR IF FINISHED WITH COMMAND
004732  120027  000054                CMPB    R0,#',          ;DID HE SPECIFY NBPC?
004736  001006                        BNE     2$              ;BR IF NOT
004740  004767  000000G                CALL    ACRDEC          ;ACCRUE NUMBER OF BYTES PER CELL
004744  010167  000000G                MOV     R1,MNBPC
004750  112300                4$:     MOVB    (R3)+,R0        ;GET DELIMITER
004752  001417                        BEQ     1$              ;BR IF END OF COMMAND HIT
004754  120027  000057        2$:     CMPB    R0,#'/          ;GOT A SWITCH?
004760  001053                        BNE     3$              ;BR IF NOT
004762  012704  000234'               MOV     #MONHD,R4       ;POINT TO TABLE OF SWITCH OPTIONS
004766  004767  000000G                CALL    SEARCH          ;SEARCH FOR SWITCH IN TABLE
004772  103004                        BCC     5$              ;BR IF FOUND IT
004774                                FABORT  #INVOPT         ;INVALID SWITCH
005004  051467  000000G        5$:     BIS     (R4),MNFLGS     ;SET BITS FOR SWITCH
005010  000757                        BR      4$              ;GO CHECK FOR MORE SWITCHES
                                ;
                                ;  Finished scanning command.
                                ;  Initiate performance monitoring.
                                ;
005012  012700  000000G        1$:     MOV     #MONAR1,R0      ;POINT TO EMT ARG BLOCK
005016  104375                        EMT     375             ;TRY TO INITIATE MONITORING
005020  103026                        BCC     6$              ;BR IF OK
005022  105737  000000G                TSTB    @#ERRLOC        ;CHECK ERROR CODE
005026  001404                        BEQ     7$              ;BR IF SOMEONE ELSE DOING MONITORING NOW
005030                                FABORT  #NOPMGN         ;PM NOT GENNED IN
005040  010046                7$:     MOV     R0,-(SP)        ;SAVE # OF PM USER
005042                                FERR    #PMBUSY         ;PM FACILITY IN USE BY ANOTHER USER
005056  012605                        MOV     (SP)+,R5        ;GET # OF PM USER
005060  004767  000000G                CALL    PRTDEC          ;PRINT DECIMAL VALUE
005064                                .PRINT  #CRLF           ;TERMINATE THE ERROR LINE
005072  000167  000000G                JMP     RDCMD
005076  012700  000000G        6$:     MOV     #MONAR2,R0      ;START MONITORING
005102  104375                        EMT     375
005104  000167  000000G                JMP     RDCMD
                                ;
                                ;  Invalid command syntax.
                                ;
005110                          3$:     FABORT  #CSIMSI
```

```
                                        .SBTTL  SPOOL command
                                ;-------------------------------------------------------
                                ;  THE SPOOL COMMAND IS USED TO CONTROL THE OPERATION
                                ;  OF SPOOLED DEVICES.
                                ;  THE FORM OF THE COMMAND IS:
                                ;    SPOOL DEV,OPTION,OPTION
                                ;
005120  004767  0000000         CMDSPO: CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACE
005124  004767  0000000                 CALL    CVTUC           ;Convert lower case chars to upper case
005130  105713                         TSTB    (R3)            ;ANY DEVICE NAMED?
005132  001002                         BNE     1$              ;BRANCH IF YES
005134  000167  0000000                 JMP     BADCMD          ;MUST HAVE DEVICE NAME
                                ;
                                ;  ACCRUE DEVICE NAME IN RAD50 FORM.
                                ;
005140  004767  0000000         1$:     CALL    GTRD50          ;ACCRUE THE NAME
                                ;
                                ;  If device name is a logical name, translate to physical
                                ;
005144  016700  0000000                 MOV     R50BUF,R0       ;Get device name
005150  004767  0000000                 CALL    ASNSRC          ;See if this name is assigned
005154  103403                          BCS     3$              ;Br if name is not assigned
005156  016267  0000000 0000000         MOV     AT$DEV(R2),R50BUF;Set physical device name
                                ;
                                ;  NOW TRY TO FIND SDCB WHICH MATCHES NAME.
                                ;
005164  010346                 3$:     MOV     R3,-(SP)        ;SAVE POINTER
005166  012703  0000000                 MOV     #R50BUF,R3      ;POINT TO DEVICE NAME
005172  004767  0000000                 CALL    FORCE0          ;MAKE UNIT 0 IF NEEDED
005176  012603                          MOV     (SP)+,R3        ;RESTORE POINTER
005200  012701  0000000                 MOV     #SDCB,R1        ;POINT TO 1ST SDCB
005204  020127  0000000         4$:     CMP     R1,#SDCBND      ;CHECKED ALL?
005210  103013                          BHIS    2$              ;BRANCH IF YES
005212  026127  0000000 0000000         CMP     SDNAME(R1),#DMYDEV;Is this dummy device entry?
005220  001404                          BEQ     5$              ;Br if yes
005222  026761  0000000 0000000         CMP     R50BUF,SDNAME(R1) ;DO NAMES MATCH?
005230  001407                          BEQ     CSGTOP          ;BRANCH IF FOUND SDCB
005232  062701  0000000         5$:     ADD     #SDCBSZ,R1      ;GO CHECK NEXT SDCB
005236  000762                          BR      4$
                                ;
                                ;  CAN'T FIND SDCB FOR DEVICE.
                                ;
005240                          2$:     FABORT  #INVDEV
                                ;
                                ;  FOUND SDCB (R1).
                                ;  IDENTIFY OPTION.
                                ;
005250  112300                 CSGTOP: MOVB    (R3)+,R0        ;SKIP OVER ANY DELIMITERS
005252  004767  0000000                 CALL    CHKDLM
005256  103774                          BCS     CSGTOP
005260  005303                          DEC     R3
005262  012704  000260'                 MOV     #SPLHD,R4       ;POINT TO TABLE OF SPOOL OPTIONS
005266  004767  0000000                 CALL    SEARCH          ;LOOK UP OPTION
005272  103401                          BCS     1$              ;BR IF ERROR
                                ;
                                ;  FOUND OPTION.  BRANCH OFF TO OPTION PROCESSING ROUTINE.
                                ;  SDCB ADDRESS IS IN R1.
```

```
                                ;
005274  000134                          JMP     @(R4)+
                                ;
                                ;  ERROR IN SEARCH
                                ;
005276  005704                  1$:     TST     R4              ;ILLEGAL OR AMBIGUOUS
005300  001404                          BEQ     2$              ;BR IF ILLEGAL
005302                                  FABORT  #AMBOPT         ;AMBIGUOUS OPTION
005312                          2$:     FABORT  #INVOPT         ;ILLEGAL OPTION
```

```
                                ;.....................................................
                                ;
                                ;  PROCESS THE 'DEL' (DELETE) OPTION.
                                ;
005322                          SPLDEL::
                                ;
                                ;  See if a file ID was specified
                                ;
005322  105713                          TSTB    (R3)            ;Was an id specified?
005324  001425                          BEQ     SPDLCR          ;Br if not
005326  121327  000054                  CMPB    (R3),#',        ;Comma used as delimiter?
005332  001403                          BEQ     1$              ;Br if yes
005334  121327  000075                  CMPB    (R3),#'=        ;Equal sign used as delimiter?
005340  001001                          BNE     2$              ;Br if not -- space must have been delimiter
005342  005203                  1$:     INC     R3              ;Skip over comma or equal sign
                                ;
                                ;  A file id was specified, accrue the id value
                                ;
005344  004767  0000000         2$:     CALL    ACRDEC          ;Accrue the value
005350  010167  172470                  MOV     R1,DELEMT+4     ;Set file ID in EMT arg block
005354  012700  000040'                MOV     #DELEMT,R0      ;Point to EMT arg block
005360  104375                          EMT     375             ;Try to delete the file
005362  103004                          BCC     9$              ;Br if successful
005364                                  FABORT  #EM$NID         ;Cannot find file with that id
005374  000167  0000000         9$:     JMP     RDCMD
                                ;
                                ;  Abort the file currently being printed by the spooled device
                                ;
005400  005761  0000000         SPDLCR: TST     SDSFCB(R1)      ;IS DEVICE BUSY?
005404  001405                          BEQ     SPNBSY          ;BRANCH IF NOT
005406  152761  0000000 0000000         BISB    #SD$DEL,SDFLAG(R1) ;SET DELETE FLAG
005414  000167  0000000         CSABT:  JMP     RDCMD
005420                          SPNBSY: .PRINT  #DEVIDL
005426  000772                          BR      CSABT
```

```
                                ;-------------------------------------------------------------------------------
                                ;  PROCESS THE SPOOL 'ALIGN' OPTION.
                                ;  (SPOOL XX,ALIGN,<FILE NAME>)
                                ;
005430  004767  0000000        SPLALN: CALL    CKPRIV          ;Must have operator privilege
005434  122327  000054                CMPB    (R3)+,#',       ;WAS FILE NAME GIVEN?
005440  001404                        BEQ     1$              ;BR IF YES
005442                          9$:     FABORT  #COAL
                                ;  PARSE ALIGNMENT FILE NAME
005452  010605                  1$:     MOV     SP,R5           ;SAVE SP
005454                                  .CSIGEN #ALDEX,#ALDEX,R3;OPEN ALIGNEMENT FILE - CHAN 3
005470  010506                        MOV     R5,SP           ;RESET SP
005472  103763                        BCS     9$              ;BRANCH IF .CSIGEN ERROR
                                ;  ALIGNEMENT FILE IS NOW OPEN ON CHANNEL 3
                                ;  OPEN SPOOLED DEVICE ON CHANNEL 1
                                ;
005474  016167  0000000 0000000        MOV     SDNAME(R1),ALDBLK;SET UP DEVICE NAME
005502                                  .ENTER  #XAREA,#1,#ALDBLK,#0;OPEN SPOOLED DEVICE
005526  103010                        BCC     2$              ;BR IF OPEN OK
005530                                  .PRINT  #COAD           ;CAN'T OPEN SPOOLED DEVICE
005536                                  .CLOSE  #3              ;CLOSE ALIGNMENT FILE
005544  000167  0000000          10$:    JMP     RDCMD
                                ;
                                ;  WRITE 1ST RECORD TO SPOOLED FILE AND SPECIFY A
                                ;  FORM NAME OF "******" WHICH MEANS ALIGNMENT FILE.
                                ;
005550                          2$:     .WRITW  #XAREA,#1,#ALFN,#4,#0
                                ;
                                ;  NOW COPY ALIGNMENT FILE TO SPOOLED FILE
                                ;
005606  005007                          CLR     R2              ;INIT BLOCK #
005610                          4$:     .READW  #XAREA,#3,#BLK0,#256.,R2
005646  103421                        BCS     3$              ;BR IF END OF FILE
005650                                  .WRITW  #XAREA,#1,#BLK0,#256.,R2
005706  005202                        INC     R2
005710  000737                        BR      4$
                                ;  HIT END OF ALIGNMENT FILE.   CLOSE BOTH CHANNELS.
005712                          3$:     .CLOSE  #1
005720                                  .CLOSE  #3
005726  000706                        BR      10$
                                ;
                                ;  PROCESS THE 'LOCK' OPTION.
                                ;
005730  004767  0000000        SPLLK:  CALL    CKPRIV          ;Must have operator privilege
005734  052761  0000000 0000000        BIS     #SD$FLK,SDFLAG(R1);SAY FORM IS LOCKED
005742  000405                        BR      SPLMT1          ;NOW DO MOUNT
                                ;
                                ;  PROCESS THE 'FORM' OPTION
                                ;
005744  004767  0000000        SPLFRM: CALL    CKPRIV          ;Must have operator privilege
005750  042761  0000000 0000000        BIC     #SD$FLK,SDFLAG(R1);SAY FORM IS NOT LOCKED
005756  010102                SPLMT1: MOV     R1,R2           ;POINT TO FORM NAME CELL
005760  062702  0000000                ADD     #SDFORM,R2
005764  010204                        MOV     R2,R4           ;POINT TO END OF CELL
005766  062704  000006                ADD     #6,R4
005772  005203                        INC     R3              ;SKIP OVER COMMA
005774  105713                  3$:   TSTB    (R3)            ;HIT END OF NAME?
```

```
      58 005776  001404                          BEQ     1$                  ;BR IF YES
      59 006000  020204                          CMP     R2,R4               ;COPIED 6 CHARS?
      60 006002  103007                          BHIS    2$                  ;BR IF YES
      61 006004  112322                          MOVB    (R3)+,(R2)+         ;COPY IN FORM NAME
      62 006006  000772                          BR      3$
      63                                ;  PAD END OF SHORT NAME WITH BLANKS
      64 006010  020204                 1$:      CMP     R2,R4               ;FILLED TO 6 CHARS?
      65 006012  103003                          BHIS    2$                  ;BR IF YES
      66 006014  112722  000040                  MOVB    #' ,(R2)+           ;PUT IN TRAILING SPACES
      67 006020  000773                          BR      1$
      68                                ;
      69                                ;  NOW TRY TO START SPOOLER
      70                                ;
      71 006022  042761  0000000 0000000 2$:      BIC     #SD$WFM,SDFLAG(R1);SAY FORM MOUNT IS DONE
      72 006030  010100                          MOV     R1,R0               ;POINT TO SDCB
      73 006032  004767  000464                  CALL    SPOLGO              ;TRY TO START SPOOLER
      74 006036  000167  0000000                 JMP     RDCMD
      75                                ;
      76                                ;  PROCESS THE 'SKIP' OPTION
      77                                ;
      78 006042  004767  0000000        SPLSKP: CALL    CKPRIV              ;Must have operator privilege
      79 006046  005761  0000000                 TST     SDSFCB(R1)          ;IS DEVICE BUSY?
      80 006052  001415                          BEQ     SPNBB               ;BR IF NOT
      81                                ; ACCRUE NUMBER
      82 006054  010104                          MOV     R1,R4               ;SAVE SDCB ADDRESS
      83 006056  005203                          INC     R3                  ;SKIP OVER COMMA
      84 006060  004767  0000000                 CALL    ACRDEC              ;ACCRUE DEC NUMBER
      85 006064  010164  0000000                 MOV     R1,SDSKIP(R4)       ;SET SKIP COUNT IN SDCB
      86 006070  000167  0000000                 JMP     RDCMD
      87                                ;
      88                                ;  PROCESS THE 'BACK' OPTION
      89                                ;
      90 006074  004767  0000000        SPLBAK: CALL    CKPRIV              ;Must have operator privilege
      91 006100  005761  0000000                 TST     SDSFCB(R1)          ;IS THE DEVICE BUSY
      92 006104  001002                          BNE     SPLBK1              ;BR IF YES
      93 006106  000167  177306         SPNBB:  JMP     SPNBSY              ;GO GIVE ERROR MESSAGE
      94 006112  005761  0000000        SPLBK1: TST     SDBU(R1)            ;ANY BLOCKS REMEMBERED YET?
      95 006116  001403                          BEQ     1$                  ;BR IF NOTHING TO BACKUP TO
      96 006120  052761  0000000 0000000       BIS     #SD$BAK,SDFLAG(R1);REQUEST BACK UP
      97 006126  000167  0000000        1$:     JMP     RDCMD
```

```
                                        ;
                                        ;  PROCESS THE SPOOL 'STATUS' COMMAND
                                        ;
006132  005761  000000    SPLSTA: TST     SDSFCB(R1)          ;IS SPOOLER BUSY?
006136  001404                    BEQ     1$                  ;BR IF NOT
006140                            .PRINT  #SPACTV
006146  000413                    BR      4$
006150  032761  000000  000000  1$:       BIT     #SD$WFM,SDFLAG(R1);IS IT WAITING FOR A FORM MOUNT?
006156  001404                    BEQ     2$
006160                            .PRINT  #SPWFM
006166  000403                    BR      4$
006170                          2$:       .PRINT  #DEVIDL
                                        ;  SEE IF IN SINGLE FILE MODE
006176  032761  000000  000000  4$:       BIT     #SD$SNG,SDFLAG(R1);IN SINGLE-FILE MODE?
006204  001403                    BEQ     20$                 ;BR IF NOT
006206                            .PRINT  #SPSNG
                                        ;  SEE IF SPOOL FILE IS FULL
006214  005767  000000          20$:    TST     NFRESB              ;IS SPOOL FILE FULL?
006220  001003                    BNE     3$                  ;BR IF NOT
006222                            .PRINT  #SPFUL
                                        ;  LIST CURRENT FORM NAME
006230                          3$:       .PRINT  #SPCF
006236  010102                    MOV     R1,R2               ;POINT TO FORM NAME CELL
006240  062702  000000            ADD     #SDFORM,R2
006244  012703  000006            MOV     #6,R3               ;PRINT 6 CHARS
006250  112200                  5$:       MOVB    (R2)+,R0
006252                            .TTYOUT
006256  005303                    DEC     R3
006260  001373                    BNE     5$
                                        ;  SEE IF CURRENT FORM IS LOCKED
006262  032761  000000  000000            BIT     #SD$FLK,SDFLAG(R1);IS FORM LOCKED?
006270  001404                    BEQ     6$                  ;BR IF NOT
006272                            .PRINT  #SPFLK
006300  000403                    BR      7$
006302                          6$:       .PRINT  #CRLF
                                        ;  List info about files pending for this device.
006310  005761  000000          7$:     TST     SDFHD(R1)           ;ARE THERE ANY FILES PENDING FOR THIS DEVICE?
006314  001004                    BNE     11$                 ;BR IF YES
006316                            .PRINT  #NOFIL              ;NO FILES...
006324  000410                    BR      8$
006326                          11$:      .PRINT  #QHDMS1             ;PRINT HEADING MESSAGE
006334                            .PRINT  #QHDMS2             ;UNDERLINE THE HEADING
006342  004767  000000            CALL    LSTSPL              ;LIST SPOOL FILES
006346  000167  000000          8$:     JMP     RDCMD
                                        ;
                                        ;  PROCESS THE 'SINGLE' OPTION
                                        ;
006352  004767  000000  SPLSNG: CALL     CKPRIV              ;Must have operator privilege
006356  052761  000000  000000        BIS     #SD$SNG,SDFLAG(R1);SET SINGLE-FILE MODE
006364  000167  000000            JMP     RDCMD
                                        ;
                                        ;  PROCESS THE 'MULTIPLE' OPTION
                                        ;
006370  004767  000000  SPLMUL: CALL     CKPRIV              ;Must have operator privilege
006374  042761  000000  000000        BIC     #SD$SNG,SDFLAG(R1);RESET SINGLE-FILE MODE
006402  000167  000000            JMP     RDCMD
                                        ;
```

```
                                ;  PROCESS THE SPOOL 'HOLD' COMMAND.
                                ;  HOLD MEANS DON'T START OUTPUT UNTIL CHANNEL IS CLOSED.
                                ;
006406  004767  0000000        SPLHLD: CALL    CKPRIV          ;Must have operator privilege
006412  010167  171434                MOV     R1,SPHLEM+4     ;Set SDCB address in EMT arg block
006416  012700  000046'              MOV     #SPHLEM,R0      ;Point to EMT arg block
006422  104375                      EMT     375             ;Set hold mode
006424  000167  0000000              JMP     RDCMD
                                ;
                                ;  PROCESS THE SPOOL 'NOHOLD' COMMAND WHICH MEANS START OUTPUT
                                ;  AS SOON AS A FILE IS CREATED.
                                ;
006430  004767  0000000        SPLNHL: CALL    CKPRIV          ;Must have operator privilege
006434  010167  171420                MOV     R1,SPNHEM+4     ;Set SDCB address in EMT arg block
006440  012700  000054'              MOV     #SPNHEM,R0      ;Point to EMT arg block
006444  104375                      EMT     375             ;Set NOHOLD mode
006446  000167  0000000              JMP     RDCMD
                                ;
                                ;-----------------------------------------------------
                                ;  PROCESS THE 'FORM' COMMAND.
                                ;
006452  004767  0000000        CMDFRM: CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACE
006456  004767  0000000              CALL    CVTUC           ;CONVERT ALL CHARS TO UPPER CASE
006462  012704  0000000              MOV     #UFORM,R4       ;POINT TO CELL WHICH HOLDS FORM NAME
006466  020302                  2$: CMP     R3,R2           ;MOVED ALL OF NAME?
006470  103004                      BHIS    1$              ;BR IF YES
006472  112324                      MOVB    (R3)+,(R4)+     ;MOVE FORM NAME TO UFORM
006474  020427  0000060            CMP     R4,#<UFORM+6>   ;MOVED 6 CHARS?
006500  103772                      BLO     2$              ;BR IF NOT
                                ;  SEE IF WE NEED TO PAD NAME WITH BLANKS
006502  020427  0000060        1$: CMP     R4,#<UFORM+6>   ;FILLED TO 6 CHARS?
006506  103003                      BHIS    3$              ;BR IF FINISHED
006510  112724  000040              MOVB    #' ,(R4)+       ;PUT IN TRAILING BLANKS
006514  000772                      BR      1$
006516  000167  0000000          3$: JMP     RDCMD

                                ;-------------------------------------------------------------------------
                                ;  SPOLGO is called to start a spooler.
                                ;
                                ;  Inputs:
                                ;   R0 = SDCB address of spooler to be started.
                                ;
006522  010046                  SPOLGO: MOV     R0,-(SP)
006524  010067  0000040              MOV     R0,SPGEMT+4     ;SET SDCB ADDRESS IN EMT ARG BLOCK
006530  012700  0000000              MOV     #SPGEMT,R0      ;POINT TO ARG BLOCK
006534  104375                      EMT     375             ;START THE SPOOLER
006536  012600                      MOV     (SP)+,R0
006540  000207                      RETURN
```

```
                                        .SBTTL  SEND command
                                ;-------------------------------------------------------------------------
                                ;  Process the YELL command which overrides TT gag but requires
                                ;  operator privilege.
                                ;
006542  004767  0000000         CMDYEL: CALL    CKPRIV          ;Require OPER privilege
006546  112767  000003  0000000        MOVB    #3,YELEMT       ;Set the YELL and error flags
006554  000411                        BR      SNDCOM          ;Enter SEND command code

                                ;-------------------------------------------------------
                                ;  Process the OPERATOR command which functions as a send
                                ;  command to the operators console.
                                ;
006556  116701  0000000         OPRCMD: MOVB    CTRLTT,R1       ;GET LINE INDEX # OF OPR TERMINAL
006562  001014                        BNE     SNDON           ;BR IF GOT ONE
006564                                FABORT  #NOOPTT         ;NO OPERATOR'S CONSOLE

                                ;-------------------------------------------------------
                                ;  PROCESS THE 'SEND' COMMAND.
                                ;
006574  105067  0000000         CMDSND: CLRB    YELEMT          ;This is not the YELL command
006600  111300                  SNDCOM: MOVB    (R3),R0         ;GET 1ST CHAR OF COMMAND
006602  001420                        BEQ     CSEXIT          ;BRANCH IF NULL COMMAND
                                ;
                                ;  ACCRUE LINE #.
                                ;
006604  004767  000544                CALL    SNDLIN          ;Accrue line # to send to
006610  005701                        TST     R1              ;SEND TO SPECIFIC USER?
006612  003422                        BLE     CSSCAN          ;BRANCH IF ALL OR OC
006614  020127  0000000       SNDON:  CMP     R1,#LSTSL       ;IS THIS A VALID LINE #?
006620  101003                        BHI     LNUMER          ;BR IF NOT
006622  016101  0000000               MOV     LNPRIM(R1),R1   ;GET PRIMARY LINE #
006626  001010                        BNE     CSCD            ;BRANCH IF VALID
006630                          LNUMER: FWARN   #BDLIN          ;Invalid line number
006644  000167  0000000         CSEXIT: JMP     RDCMD
006650  032761  0000000 0000000 CSCD:   BIT     #$DETCH,LSW(R1) ;TRYING TO SEND TO DETACHED JOB?
006656  001364                        BNE     LNUMER          ;BR IF YES
                                ;
                                ;  AT THIS POINT THE INDEX NUMBER OF THE LINE WE WISH TO
                                ;  SEND TO IS IN R1.
                                ;  PUT OUR LINE NUMBER AND USER NAME AT START OF MESSAGE BUFFER
                                ;
006660  032767  0000000 0000000 CSSCAN: BIT     #PO$SND,PRIVCO  ;Are we privileged to send messages?
006666  001004                        BNE     10$             ;Br if yes
006670                                FABORT  #EM$SND
006700  012704  0000000         10$:    MOV     #MSGBUF,R4      ;POINT TO MESSAGE BUFFER
006704  112724  000015                 MOVB    #CR,(R4)+       ;PUT IN LEADING CR-LF-BELL
006710  112724  000012                 MOVB    #LF,(R4)+
006714  112724  000007                 MOVB    #BELL,(R4)+
006720  116705  0000000               MOVB    CORUSR,R5       ;GET OUR LINE NUMBER
006724  016505  0000000               MOV     LNPRIM(R5),R5   ;GET OUR PRIMARY LINE NUMBER
006730  010502                        MOV     R5,R2
006732  062702  0000000               ADD     #NUMTB1,R2      ;POINT TO TABLE WITH LINE NUMBER CHARACTERS
006736  112224                        MOVB    (R2)+,(R4)+     ;PUT IN LINE NUMBER
006740  112224                        MOVB    (R2)+,(R4)+
006742  070527  000006              MUL     #6,R5           ;EACH JOB HAS A 12 CHAR USER NAME
006746  062705  0000000               ADD     #LUNAME,R5      ;POINT TO USER NAME ENTRY FOR THIS LINE
```

```
     58 006752  121527  000040                    CMPB    (R5),#'         ;IS USER NAME BLANK?
     59 006756  001416                            BEQ     4$              ;BR IF YES
     60 006760  112724  000040                    MOVB    #' ,(R4)+       ;PUT IN A SPACE
     61 006764  112724  000050                    MOVB    #'(,(R4)+       ;AND OPEN PAREN AT START OF NAME
     62 006770  012702  000014                    MOV     #12.,R2         ;GET LENGTH OF NAME
     63 006774  112524                  6$:       MOVB    (R5)+,(R4)+     ;GET NEXT CHAR OF USER NAME
     64 006776  077202                            SOB     R2,6$           ;MOVE REST OF NAME
     65 007000  124427  000040          7$:       CMPB    -(R4),#BLANK    ;TRIM OFF TRAILING SPACES
     66 007004  001775                            BEQ     7$
     67 007006  005204                            INC     R4              ;POINT BEYOND LAST NON-BLANK CHARACTER
     68 007010  112724  000051          5$:       MOVB    #'),(R4)+       ;PUT IN CLOSE PAREN AT END OF NAME
     69 007014  112724  000040          4$:       MOVB    #' ,(R4)+       ;PUT " -- " AT END OF NAME
     70 007020  112724  000055                    MOVB    #'-,(R4)+
     71 007024  112724  000055                    MOVB    #'-,(R4)+
     72 007030  112724  000040                    MOVB    #' ,(R4)+
     73                                  ;
     74                                  ;  Now move message string to message buffer
     75                                  ;
     76 007034  122327  000040          3$:       CMPB    (R3)+,#'        ;SKIP OVER ANY BLANKS
     77 007040  001775                            BEQ     3$
     78 007042  005303                            DEC     R3              ;POINT TO 1ST CHAR OF MESSAGE
     79 007044  112324                  1$:       MOVB    (R3)+,(R4)+     ;MOVE MESSAGE TO OUR BUFFER
     80 007046  001403                            BEQ     2$              ;BRANCH WHEN END HIT
     81 007050  020427  177760                    CMP     R4,#<MSGEND-2>  ;DON'T OVERFLOW OUR BUFFER
     82 007054  103773                            BLO     1$
     83 007056  005304                  2$:       DEC     R4              ;PUT CR-LF AT END OF MESSAGE
     84 007060  112724  000015                    MOVB    #CR,(R4)+
     85 007064  112724  000012                    MOVB    #LF,(R4)+
     86 007070  105014                            CLRB    (R4)            ;PUT IN NULL TO SIGNAL END
     87                                  ;
     88                                  ;  See who we should send the message to.
     89                                  ;
     90 007072  005701                            TST     R1              ;WHO IS IT GOING TO?
     91 007074  001655                            BEQ     LNUMER          ;SEND TO CONSOLE TERMINAL
     92 007076  100450                            BMI     SNDALL          ;SEND TO ALL LINES
     93                                  ;
     94                                  ;  Send message to one terminal whose line index number
     95                                  ;  is in R1.
     96                                  ;
     97 007100  032761  000000  000000  SNDONE:  BIT     #$DILUP,LSW(R1) ;IS THAT LINE LOGGED ON?
     98 007106  001005                            BNE     1$              ;BRANCH IF YES
     99 007110                                    .PRINT  #NOTON
    100 007116  000167  000000                    JMP     RDCMD
    101 007122  032761  000000  000000  1$:       BIT     #$TTGAG,LSW7(R1);IS LINE GAGGED?
    102 007130  001414                            BEQ     2$              ;BR IF NOT
    103 007132  032761  000000  000000            BIT     #$INKMN,LSW4(R1);IS JOB IN KMON?
    104 007140  001010                            BNE     2$              ;BR IF YES
    105 007142  105767  000000                    TSTB    YELEMT          ;Are we yelling?
    106 007146  001005                            BNE     2$              ;Br if yes -- Override gag
    107 007150                                    .PRINT  #GAGMSG         ;SAY LINE IS GAGGED
    108 007156  000167  000000                    JMP     RDCMD
    109 007162  012700  000000          2$:       MOV     #YELEMT,R0      ;POINT TO EMT ARG BLOCK
    110 007166  006201                            ASR     R1              ;GET LINE NUMBER * 1
    111 007170  010160  000002                    MOV     R1,2(R0)        ;SET LINE # IN ARG BLOCK
    112 007174  104375                            EMT     375             ;SEND THE MESSAGE
    113 007176  103006                            BCC     3$              ;BR IF NO ERROR
    114 007200                                    FWARN   #NMSGDF
```

```
     115 007214  000167  000000G         3$:     JMP     RDCMD
     116                                 ;
     117                                 ;  Send message to all terminals.
     118                                 ;
     119 007220  012703  000000G         SNDALL: MOV     #LSTPL,R3       ;GET INDEX # OF LAST LINE
     120 007224  116702  000000G                 MOVB    CORUSR,R2       ;GET OUR LINE INDEX #
     121 007230  016202  000000G                 MOV     LNPRIM(R2),R2   ;GET OUR PRIMARY LINE INDEX #
     122 007234  032763  000000G 000000G 2$:    BIT     #$DILUP,LSW(R3) ;IS THIS LINE LOGGED ON?
     123 007242  001437                          BEQ     1$              ;BRANCH IF NOT
     124 007244  032763  000000G 000000G        BIT     #$DETCH,LSW(R3) ;IS THIS A DETACH LINE?
     125 007252  001033                          BNE     1$              ;BR - DETACH LINES DON'T NEED MESSAGES
     126 007254  020302                          CMP     R3,R2           ;DON'T SEND MESSAGE TO OURSELF
     127 007256  001431                          BEQ     1$
     128 007260  032763  000000G 000000G        BIT     #$TTGAG,LSW7(R3);IS LINE GAGGED?
     129 007266  001407                          BEQ     3$              ;BR IF NOT
     130 007270  105767  000000G                 TSTB    YELEMT          ;Are we YELLing?
     131 007274  001004                          BNE     3$              ;Br if yes -- Override gag
     132 007276  032763  000000G 000000G        BIT     #$INKMN,LSW4(R3);IS JOB IN KMON NOW?
     133 007304  001416                          BEQ     1$              ;IF NOT THEN DON'T SEND MESSAGE
     134 007306  012700  000000G         3$:     MOV     #YELEMT,R0      ;POINT TO EMT ARG BLOCK
     135 007312  010301                          MOV     R3,R1           ;GET LINE INDEX #
     136 007314  006201                          ASR     R1              ;GET LINE NUMBER
     137 007316  010160  000002                  MOV     R1,2(R0)        ;STORE LINE # IN EMT ARG BLOCK
     138 007322  104375                          EMT     375             ;SEND THE MESSAGE
     139 007324  103006                          BCC     1$              ;BR IF NO ERROR
     140 007326                                  FWARN   #NMSGBF         ;REPORT NO MESSAGE BUFFERS
     141 007342  162703  000002          1$:     SUB     #2,R3           ;MOVE ON TO NEXT LINE
     142 007346  001332                          BNE     2$              ;BRANCH IF MORE TO DO
     143 007350  000167  000000G                 JMP     RDCMD
```

```
                                ;-----------------------------------------------------------------------
                                ;  Accrue the line number associated with a SEND command
                                ;
                                ;  Inputs:
                                ;   R3 = Points past end of SEND keyword.
                                ;
                                ;  Outputs:
                                ;   R1 = Index number of job to send to (-1 ==> Send to all jobs)
                                ;
007354  010546                  SNDLIN: MOV     R5,-(SP)
                                ;
                                ;  See if a line number was specified
                                ;
007356  004767  0000000                 CALL    SKPSPC          ;Skip over any spaces
007362  121327  000054                  CMPB    (R3),#',        ;Start of line # field?
007366  001407                          BEQ     1$              ;Br if yes
007370  121327  000057                  CMPB    (R3),#'/        ;Start of line # field?
007374  001404                          BEQ     1$              ;Br if yes
007376                                  FABORT  #EM$NLN         ;No line number was specified
                                ;
                                ;  See if we are sending to ALL, OPERATOR, or PARENT
                                ;
007406  005203                  1$:     INC     R3              ;Point past delimiter
007410  004767  0000000                 CALL    SKPSPC          ;Scan up to line number
007414  111300                          MOVB    (R3),R0         ;Get 1st char of "number"
007416  020027  000141                  CMP     R0,#141         ;Is this a lower-case letter?
007422  103402                          BLO     2$              ;Br if not
007424  162700  000040                  SUB     #40,R0          ;Convert lower-case to upper case
007430  120027  000101          2$:     CMPB    R0,#'A          ;Send to ALL?
007434  001003                          BNE     3$              ;Br if not
007436  012701  177777                  MOV     #-1,R1          ;Line # for All
007442  000424                          BR      7$
007444  120027  000117          3$:     CMPB    R0,#'O          ;Send to operator?
007450  001007                          BNE     4$              ;Br if not
007452  116701  0000000                 MOVB    CTRLTT,R1       ;Get operator line #
007456  001016                          BNE     7$
007460                                  FABORT  #NOOPTT         ;No operator terminal
007470  120027  000120          4$:     CMPB    R0,#'P          ;Send to parent job?
007474  001015                          BNE     5$              ;Br if not
007476  116701  0000000                 MOVB    CORUSR,R1       ;Get current job index #
007502  005761  0000000                 TST     LPARNT(R1)      ;Is there a parent job?
007506  001402                          BEQ     7$              ;Br if yes
007510  016101  0000000                 MOV     LPARNT(R1),R1   ;Get # of parent job
007514  112300                  7$:     MOVB    (R3)+,R0        ;Skip over rest of keyword
007516  001407                          BEQ     9$
007520  120027  000040                  CMPB    R0,#'           ;Space?
007524  001373                          BNE     7$
007526  000403                          BR      9$
                                ;
                                ;  We must have an ordinary line number
                                ;
007530  004767  0000000         5$:     CALL    ACRDEC          ;Accrue line number
007534  006301                          ASL     R1              ;Convert to line index number
                                ;
                                ;  Finished
                                ;
007536  012605                  9$:     MOV     (SP)+,R5
```

SEND command

```
     58 007540  000207                          RETURN
```

```
                                        .SBTTL  ACCESS command
                                ;-------------------------------------------------------------
                                ;  PROCESS THE ACCESS COMMAND
                                ;
007542  032761  0000006 0000006 CMDACC: BIT     #$SUCF,LSW9(R1) ;ARE WE INSIDE INITIAL COMMAND FILE?
007550  001010                          BNE     15$             ;BR IF YES
007552  032767  0000006 0000006         BIT     #PO$SYS,PRIVCO  ;Does user have SYSPRV privilege?
007560  001004                          BNE     15$             ;Br if yes
007562                                  FABORT  #EM$NSF         ;ONLY LEGAL IN START-UP COMMAND FILES
007572  004767  0000006         15$:    CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACES
007576  012704  0000006                 MOV     #OKFILE,R4      ;POINT TO TABLE OF FILE NAMES
007602  105713                  3$:     TSTB    (R3)            ;REACHED END OF COMMAND LINE?
007604  001002                          BNE     1$              ;BR IF NOT
007606  000167  0000006                 JMP     RDCMD           ;FINISHED COMMAND
007612  020427  0000006         1$:     CMP     R4,#OKFEND      ;TABLE OVERFLOW?
007616  103404                          BLO     16$             ;BR IF NOT
007620                                  FABORT  #TBLOVF         ;TABLE OVERFLOW
007630  005764  0000006         16$:    TST     OF$FIL(R4)      ;SEARCH FOR FREE SLOT
007634  001403                          BEQ     4$              ;BR IF FOUND ONE
007636  062704  0000006                 ADD     #OF$$SZ,R4      ;CHECK NEXT SLOT
007642  000763                          BR      1$
                                ;
                                ;  Initialize table entry
                                ;
007644  005267  0000006         4$:     INC     RESDEV          ;ANOTHER AUTHORIZED DEVICE-FILE
007650  012702  132500                  MOV     #WLDNAM,R2      ;GET WILD-CARD FILE SPEC VALUE
007654  010264  0000006                 MOV     R2,OF$FIL(R4)   ;SET FILE NAME TO WILD CARD
007660  010264  0000026                 MOV     R2,OF$FIL+2(R4)
007664  010264  0000046                 MOV     R2,OF$FIL+4(R4)
007670  112764  177777  0000006         MOVB    #-1,OF$DEV(R4)  ;WILDCARD THE DEVICE NAME TOO
007676  112764  177777  0000006         MOVB    #-1,OF$UNT(R4)
                                ;
                                ;  Accrue device and file name.
                                ;
007704  004767  0000006                 CALL    GTRD50          ;SCAN FIRST NAME
007710  121327  000072                  CMPB    (R3),#':        ;IS COLON THE TERMINATING CHARACTER?
007714  001036                          BNE     5$              ;BR IF NOT
                                ;
                                ;  We just accrued the device name.
                                ;  See if this is a logical name that needs to be converted to physical.
                                ;
007716  016700  0000006                 MOV     R50BUF,R0       ;Get specified device name
007722  020027  132500                  CMP     R0,#WLDNAM      ;Wildcard?
007726  001417                          BEQ     6$              ;Br if wildcard device name
007730  004767  0000006                 CALL    ASNSRC          ;See if name is in assign table
007734  103403                          BCS     8$              ;Br if name not in assign table
007736  016267  0000006 0000006         MOV     AT$DEV(R2),R50BUF;Replace logical dev name with physical name
                                ;
                                ;  Convert the name to a device table index number.
                                ;
007744  016701  0000006         8$:     MOV     R50BUF,R1       ;GET THE RAD50 DEVICE NAME
007750  004767  000270                  CALL    CVTDEV          ;CONVERT DEVICE NAME TO INDEX NUMBER
007754  103473                          BCS     ACBDFS          ;BR IF BAD FILE NAME
007756  110164  0000006                 MOVB    R1,OF$DEV(R4)   ;SET DEVICE INDEX NUMBER
007762  110264  0000006                 MOVB    R2,OF$UNT(R4)   ;SET UNIT NUMBER
                                ;
                                ;  Get file name.
```

```
                                        ;
007766  005203                  6$:     INC     R3                      ;SKIP PAST COLON
007770  111300                          MOVB    (R3),R0                 ;GET FIRST CHAR OF FILE NAME
007772  120027  000052                  CMPB    R0,#'*                  ;WILD CARD?
007776  001403                          BEQ     7$                      ;BR IF YES
010000  004767  0000000                 CALL    CHKDLM                  ;1ST CHAR OF NAME SHOULDN'T BE DELIMITER
010004  103426                          BCS     10$                     ;BR IF IT IS DELIMITER
010006  004767  0000000          7$:     CALL    GTRD50                  ;ACCRUE FILE NAME
010012  016700  0000000          5$:     MOV     R50BUF,R0               ;GET RAD50 FILE NAME
010016  001452                          BEQ     ACBDFS                  ;BR IF NULL NAME
010020  020027  132500                  CMP     R0,#WLDNAM              ;WILDCARD NAME?
010024  001405                          BEQ     9$                      ;BR IF YES
010026  010064  0000000                 MOV     R0,OF$FIL(R4)           ;SET FILE NAME IN FILE SPEC
010032  016764  0000020 0000020         MOV     R50BUF+2,OF$FIL+2(R4);SET 2ND HALF OF NAME
                                ;  See if extension was specified.
010040  121327  000056          9$:     CMPB    (R3),#'.                ;WAS FILE EXTENSION SPECIFIED?
010044  001006                          BNE     10$                     ;BR IF NOT
010046  005203                          INC     R3                      ;SKIP PAST PERIOD
010050  004767  0000000                 CALL    GTRD50                  ;ACCRUE THE EXTENSION
010054  016764  0000000 0000040         MOV     R50BUF,OF$FIL+4(R4);SET EXTENSION
                                ;
                                ;  See if any switch was specified with file spec.
                                ;
010062  121327  000057          10$:    CMPB    (R3),#'/                ;WAS A SWITCH SPECIFIED?
010066  001020                          BNE     11$                     ;BR IF NOT
010070  116300  000001                  MOVB    1(R3),R0                ;GET 1ST CHAR OF SWITCH NAME
010074  120027  000122                  CMPB    R0,#'R                  ;"READ" ONLY?
010100  001403                          BEQ     13$                     ;BR IF YES
010102  120027  000162                  CMPB    R0,#'r                  ;CHECK LOWER-CASE SPELLING
010106  001052                          BNE     ACBDSW                  ;BR IF INVALID SWITCH
010110  152764  0000000 0000000 13$:     BISB    #OT$RON,OF$FLG(R4);SET READ-ONLY FLAG IN FILE SPEC
010116  005203                  12$:     INC     R3                      ;SKIP OVER /
010120  111300                          MOVB    (R3),R0                 ;GET NEXT CHAR OF SWITCH NAME
010122  004767  0000000                 CALL    CHKDLM                  ;SKIP UP TO NEXT DELIMITER
010126  103373                          BCC     12$
                                ;
                                ;  Finished getting a file spec.
                                ;
010130  062704  0000000          11$:    ADD     #OF$$SZ,R4              ;POINT TO NEXT SLOT IN ACCESS TABLE
010134  004767  000236                  CALL    SKPD2                   ;SKIP OVER TERMINATING DELIMITER
010140  000167  177436                  JMP     3$                      ;GO GET NEXT FILE SPEC IF ANY
                                ;
                                ;  Error -- Invalid file spec.
                                ;
010144                          ACBDFS: FWARN   #CSIMS2                 ;Invalid device
010160  005367  0000000                 DEC     RESDEV                  ;Remove this entry from the access table
010164  005064  0000000                 CLR     OF$FIL(R4)              ;Say entry is unused
010170  105064  0000000                 CLRB    OF$DEV(R4)
010174  105064  0000000                 CLRB    OF$UNT(R4)
010200  112300                  1$:     MOVB    (R3)+,R0                ;Get next character
010202  001412                          BEQ     2$                      ;Br if hit end of command
010204  120027  000040                  CMPB    R0,#' '                 ;Is this character a space?
010210  001403                          BEQ     3$                      ;Br if yes
010212  120027  000054                  CMPB    R0,#',                  ;Is this character a comma?
010216  001370                          BNE     1$                      ;Loop if not
010220  116701  0000000          3$:     MOVB    CORUSR,R1               ;Get back job index number
010224  000167  177312                  JMP     CMDACC                  ;Go back and process the rest of the command
```

```
    115 010230  000167  000000G         2$:     JMP     RDCMD
    116                                 ;
    117                                 ;  Error -- Bad switch
    118                                 ;
    119 010234                          ACBDSW: FABORT  #INVOPT
```

```
                                ;-------------------------------------------------------------------------
                                ;  CVTDEV is called to convert a RAD50 device name to the corresponding
                                ;  device table index number and unit number.
                                ;
                                ;  Inputs:
                                ;   R1 = RAD50 device name.
                                ;
                                ;  Outputs:
                                ;   R1 = Device table index number for device.
                                ;   R2 = Unit number.
                                ;   C-flag set on return if invalid device name.
                                ;
010244                          CVTDEV:
                                ;
                                ;  Get device name and split off unit number.
                                ;
010244  005000                          CLR     R0              ;SET FOR DIVIDE
010246  071027  000050                  DIV     #50,R0          ;SPLIT OFF LOW-ORDER RAD50 CHARACTER
010252  012702  177777                  MOV     #-1,R2          ;ASSUME NO UNIT NUMBER SPECIFIED
010256  005701                          TST     R1              ;WAS A UNIT NUMBER SPECIFIED?
010260  001406                          BEQ     6$              ;BR IF NOT
010262  162701  000036                  SUB     #36,R1          ;CONVERT RAD50 DIGIT TO BINARY VALUE
010266  010102                          MOV     R1,R2           ;PUT UNIT NUMBER IN R2
010270  020227  000011                  CMP     R2,#9.          ;MAKE SURE UNIT NUMBER IN RANGE 0-9.
010274  101027                          BHI     8$              ;BR IF INVALID UNIT NUMBER
010276  070027  000050          6$:     MUL     #50,R0          ;GET DEVICE NAME WITHOUT UNIT #
                                ;
                                ;  The RAD50 device name less unit number is now in R1.
                                ;  The unit number is in R2.
                                ;
                                ;  Translate SY and DK.
                                ;
010302  020167  0000000                 CMP     R1,R50SY        ;IS DEVICE NAME SY?
010306  001403                          BEQ     2$              ;BR IF YES
010310  020167  0000000                 CMP     R1,R50DK        ;IS DEVICE NAME DK?
010314  001007                          BNE     3$              ;BR IF NOT
010316  016701  0000000         2$:     MOV     SYINDX,R1       ;GET SY DEVICE INDEX NUMBER
010322  005702                          TST     R2              ;WAS A UNIT NUMBER SPECIFIED?
010324  002016                          BGE     4$              ;BR IF YES
010326  116702  0000010                 MOVB    SYUNIT+1,R2     ;GET SY UNIT NUMBER
010332  000413                          BR      4$
                                ;
                                ;  Look up device name in permanent device name table.
                                ;
010334  016700  0000000         3$:     MOV     NUMDEV,R0       ;GET INDEX OF LAST PERM NAME
010340  020160  0000000         5$:     CMP     R1,PNAME(R0)    ;LOOK UP DEVICE NAME
010344  001405                          BEQ     7$              ;BR IF FOUND
010346  162700  000002                  SUB     #2,R0           ;CHECK NEXT ENTRY
010352  002372                          BGE     5$
                                ;
                                ;  Invalid device name
                                ;
010354  000261                  8$:     SEC                     ;SIGNAL ERROR ON RETURN
010356  000405                          BR      9$
                                ;
                                ;  Found device name.
                                ;
```

```
      58 010360  010001                  7$:     MOV     R0,R1           ;GET DEVICE NAME TABLE INDEX
      59 010362  005702                  4$:     TST     R2              ;WAS A UNIT NUMBER SPECIFIED?
      60 010364  002001                          BGE     10$             ;BR IF YES
      61 010366  005002                          CLR     R2              ;CONVERT NO UNIT NUMBER TO # 0
      62 010370  000241                  10$:    CLC                     ;SIGNAL NO ERROR ON RETURN
      63                                 ;
      64                                 ;  Return
      65                                 ;
      66 010372  000207                  9$:     RETURN
      67                                 ;-------------------------------------------------------------------------------
      68                                 ;  SUBROUTINE TO SKIP OVER COMMAS AND SPACES
      69                                 ;
      70 010374  005203                  SKPD1:  INC     R3              ;POINT TO NEXT CHAR
      71 010376  121327  000040          SKPD2:  CMPB    (R3),#' '       ;IS IT A SPACE
      72 010402  001774                          BEQ     SKPD1           ;KEEP SKIPPING IF YES
      73 010404  121327  000054                  CMPB    (R3),#','       ;IS IT A COMMA
      74 010410  001771                          BEQ     SKPD1           ;KEEP SKIPPING IF YES
      75 010412  000207                          RETURN
```

```
                                        .SBTTL  DETACH command
                                ;-------------------------------------------------------------
                                ;  PROCESS THE DETACH COMMAND
                                ;
010414  004767  0000006         CMDDET: CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACES
010420  032767  0000000 0000000         BIT     #PO$DET,PRIVCO  ;IS USER ALLOWED TO USE DETACHED JOBS?
010426  001004                         BNE     1$              ;BR IF YES
010430                                 FABORT  #EM$DET         ;NOT ALLOWED
                                ;
                                ;  SEE IF ANY SWITCHES WERE SPECIFIED WITH COMMAND
                                ;
010440  121327  000057          1$:    CMPB    (R3),#'/        ;ANY SWITCHES?
010444  001062                         BNE     DETGO           ;BR IF NOT -- START DETACHED JOB
                                ;  ACCRUE AND IDENTIFY SWITCH
010446  005203                         INC     R3              ;SKIP OVER /
010450  012704  000220'               MOV     #DETHD,R4       ;POINT TO TABLE OF SWITCHES
010454  004767  0000006               CALL    SEARCH          ;LOOK UP THE SWITCH
010460  103004                         BCC     2$              ;BR IF FOUND OK
010462                                 FABORT  #INVOPT         ;INVALID SWITCH
                                ;  ACCRUE LINE NUMBER AND SEE IF LEGAL
010472  004767  0000006         2$:    CALL    ACRDEC          ;ACCRUE THE DECIMAL NUMBER
010476  006301                         ASL     R1              ;CONVERT TO LINE INDEX #
010500  020127  0000006               CMP     R1,#FSTDL       ;SEE IF IT IS A DETACHED LINE
010504  103004                         BHIS    3$
010506                          4$:    FABORT  #BDLIN
010516  020127  0000006         3$:    CMP     R1,#LSTDL
010522  101371                         BHI     4$
010524  012700  0000006               MOV     #DETARG,R0      ;POINT TO DETACH EMT ARG BLOCK
010530  006201                         ASR     R1              ;STORE DETACHED JOB #
010532  010167  0000020               MOV     R1,DETARG+2
                                ;  BRANCH OFF TO PROCESSING ROUTINE
010536  000134                         JMP     @(R4)+          ;ENTER PROCESSING ROUTINE
                                ;
                                ;  Process the /CHECK option.
                                ;
010540  112710  000001          DETCHK: MOVB    #1,(R0)         ;SET SUB-FUNCTION CODE FOR EMT
010544  104375                         EMT     375             ;CHECK ON DETACHED JOB
010546  103404                         BCS     DETIDL          ;BR IF JOB FINISHED
010550                                 .PRINT  #RUNMS          ;JOB STILL RUNNING
010556  000403                         BR      DETJMP
010560                          DETIDL: .PRINT  #LINFRE         ;LINE IS IDLE
010566  000167  0000006        DETJMP: JMP     RDCMD           ;FINISHED WITH COMMAND
                                ;
                                ;  Process the /KILL option.
                                ;
010572  112710  000002          DETKIL: MOVB    #2,(R0)         ;SET SUB-FUNCTION CODE FOR EMT
010576  104375                         EMT     375             ;KILL A DETACHED JOB
010600  103767                         BCS     DETIDL          ;BR IF JOB NOT RUNNING
010602                                 .PRINT  #DJABMS         ;PRINT CONFIRMATION
010610  000766                         BR      DETJMP          ;FINISHED
```

```
                                        ;
                                        ;  Start a new detached job.
                                        ;  R3 = Pointer to ASCIZ string with detached job command file name.
                                        ;  Try to open the command file to make sure it exists.
                                        ;
010612  010302                  DETCO:  MOV     R3,R2           ;Save pointer to start of file name string
010614  012704  0000000                 MOV     #R50COM,R4      ;Point to word with default extension (COM)
010620  005005                          CLR     R5              ;Say this is an input file
010622  004767  0000000                 CALL    ACRFIL          ;Accrue the file spec
010626  103441                          BCS     10$             ;Br if invalid file spec
010630                                  .LOOKUP #XAREA,#1,#FILNAM ;Try to open the file
010650  103434                          BCS     11$             ;Br if cannot open file
010652                                  .CLOSE  #1              ;Close the file
                                        ;
                                        ;  Execute EMT to initiate the detached job.
                                        ;
010660  012700  0000000                 MOV     #DETARG,R0      ;POINT TO DETACH EMT ARG BLOCK
010664  105010                          CLRB    (R0)            ;SET SUB-FUNCTION CODE FOR EMT
010666  010260  000002                  MOV     R2,2(R0)        ;SET ADDRESS OF COMMAND FILE NAME ASCIZ STRING
010672  104375                          EMT     375             ;START A DETACHED JOB
010674  103412                          BCS     1$              ;BR IF NO FREE LINES
                                        ;
                                        ;  Tell user which line was used
                                        ;
010676  010005                          MOV     R0,R5           ;GET DETACHED JOB LINE NUMBER
010700                                  .PRINT  #DLMSG          ;PRINT HEADING
010706  004767  0000000                 CALL    PRTDEC          ;DISPLAY LINE #
010712                                  .PRINT  #CRLF           ;TERMINATE PRINT LINE
010720  000722                          BR      DETJMP
                                        ;
                                        ;  No free lines
                                        ;
010722                          1$:     FABORT  #NOFRDL         ;NO FREE LINES
                                        ;
                                        ;  Invalid file spec
                                        ;
010732                          10$:    FABORT  #BDFNAM
                                        ;
                                        ;  Cannot open file
                                        ;
010742                          11$:    FABORT  #EM$FUE
```

```
                                        .SBTTL  DATE command
                                ;-------------------------------------------------------------------------
                                ;  Process the DATE command.
                                ;
 010752  004767  0000000      CMDDAT: CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACES
 010756  105713                      TSTB    (R3)            ;DOES HE WANT TO SHOW OR CHANGE THE DATE?
 010760  001002                      BNE     3$              ;BR TO SET THE DATE
 010762  000167  0000000             JMP     SOPDAT          ;GO DISPLAY THE DATE
                                ;
                                ;  Set a new system date value (operator privilege required).
                                ;
 010766  004767  0000000      3$:     CALL    CKPRIV          ;MAKE SURE THIS USER IS PRIVILEGED
                                ;  Get day value.
 010772  004767  0000000             CALL    ACRDEC          ;ACCRUE DAY VALUE
 010776  005701                      TST     R1              ;CHECK RANGE OF VALUE
 011000  003453                      BLE     BADDAT          ;CAN'T BE ZERO
 011002  020127  000037              CMP     R1,#31.         ;OR GREATER THAN 31
 011006  101050                      BHI     BADDAT
 011010  072127  000005              ASH     #5,R1           ;POSITION FOR DATE WORD
 011014  010105                      MOV     R1,R5           ;CONSTRUCT DATE WORD IN R5
 011016  122327  000055              CMPB    (R3)+,#'-       ;HYPHEN SHOULD BE NEXT
 011022  001042                      BNE     BADDAT
                                ;  Get month.
 011024  004767  0000000             CALL    GTRD50          ;ACCRUE RAD50 MONTH VALUE
 011030  016700  0000000             MOV     R50BUF,R0       ;GET RAD50 VALUE OF MONTH NAME
 011034  012701  000001              MOV     #1,R1           ;FIRST MONTH IS # 1
 011040  012702  0000000             MOV     #R50MON,R2      ;POINT TO TABLE OF MONTH NAMES
 011044  020022              2$:     CMP     R0,(R2)+        ;LOOK UP NAME IN TABLE
 011046  001405                      BEQ     1$              ;BR IF FOUND
 011050  005201                      INC     R1              ;ADVANCE MONTH NUMBER
 011052  020127  000014              CMP     R1,#12.         ;ONLY 12 MONTHS
 011056  101772                      BLOS    2$
 011060  000423                      BR      BADDAT          ;BAD MONTH NAME
 011062  072127  000012      1$:     ASH     #10.,R1         ;POSITION MONTH VALUE
 011066  050105                      BIS     R1,R5           ;AND OR INTO DATE WORD
 011070  122327  000055              CMPB    (R3)+,#'-       ;SHOULD HAVE ANOTHER HYPHEN
 011074  001015                      BNE     BADDAT
                                ;  Get year
 011076  004767  0000000             CALL    ACRDEC          ;GET YEAR VALUE
 011102  020127  000143              CMP     R1,#99.         ;CHECK FOR REASONABLE UPPER LIMIT
 011106  101010                      BHI     BADDAT          ;BR IF TOO HIGH
 011110  162701  000110              SUB     #72.,R1         ;YEAR VALUE IS BIASED BY 72
 011114  003405                      BLE     BADDAT          ;BR IF TOO SMALL
 011116  050105                      BIS     R1,R5           ;FORM DATE WORD
                                ;  Set system date value.
 011120  010567  0000000             MOV     R5,SYSDAT       ;SET NEW SYSTEM DATE VALUE
 011124  000167  0000000             JMP     RDCMD           ;FINISHED WITH COMMAND
                                ;  Invalid date value.
 011130  152767  000010  0000000 BADDAT: BISB    #10,INDERR      ;SAVE ERROR LEVEL = SEVERE FOR IND
 011136                              FABORT  #INVDAT         ;INVALID DATE
```

```
                                        .SBTTL  TIME command
                                ;-------------------------------------------------------------------------------
                                ;  Process the TIME command.
                                ;
011146  004767  0000000        CMDTIM: CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACES
011152  105713                        TSTB    (R3)            ;DOES HE WANT TO SHOW OR CHANGE THE TIME?
011154  001002                        BNE     2$              ;BR TO SET THE TIME
011156  000167  0000000               JMP     SOPTIM          ;GO SHOW THE TIME
                                ;
                                ;  Set new system time (requires operator privilege)
                                ;
011162  004767  0000000        2$:     CALL    CKPRIV          ;MAKE SURE USER HAS OPERATOR PRIVILEGE
                                ;  Get hour value.
011166  004767  0000000              CALL    ACRDEC          ;ACCRUE HOUR VALUE
011172  020127  000030               CMP     R1,#24.         ;ONLY 24 HOUR IN A DAY
011176  103053                      BHIS    BADTIM
011200  010104                      MOV     R1,R4
011202  070427  000074               MUL     #60.,R4         ;CONVERT TO # MINUTES
011206  122327  000072               CMPB    (R3)+,#':       ;COLON SHOULD BE THE DELIMITER
011212  001045                      BNE     BADTIM
                                ;  Get minute value.
011214  004767  0000000              CALL    ACRDEC          ;ACCRUE MINUTE VALUE
011220  020127  000074               CMP     R1,#60.         ;ONLY 60 MINUTES PER HOUR
011224  103040                      BHIS    BADTIM
011226  060105                      ADD     R1,R5           ;COMBINE WITH HOUR VALUE
011230  005504                      ADC     R4
011232  012700  000074               MOV     #60.,R0         ;CONVERT # MINUTES TO # SECONDS
011236  004767  0000000              CALL    MUL32
                                ;  See if seconds are specified
011242  122327  000072               CMPB    (R3)+,#':       ;DELIMITER FOR SECONDS?
011246  001011                      BNE     3$              ;NO, IGNORE SECONDS
                                ;  Get second value
011250  004767  0000000              CALL    ACRDEC          ;ACCRUE SECOND VALUE
011254  020127  000074               CMP     R1,#60.         ;NO MORE THAN 60 SECONDS PER MINUTE
011260  103022                      BHIS    BADTIM
011262  060105                      ADD     R1,R5           ;COMBINE WITH HOURS AND MINUTES
011264  005504                      ADC     R4
011266  012700  000074               MOV     #60.,R0         ;60 HZ CLOCK TICKS PER SECOND
                                ;  Convert time to # clock ticks.
011272  032767  0000000  0000000 3$:     BIT     #CW$50H,CONFIG  ;50 OR 60 HZ CLOCK?
011300  001402                      BEQ     1$              ;BR IF 60 HZ
011302  012700  000062               MOV     #50.,R0         ;SET FOR 50 HZ CLOCK
011306  004767  0000000        1$:     CALL    MUL32           ;CONVERT # SECONDS TO # CLOCK TICKS
                                ;  Set new system time.
011312  010467  0000000              MOV     R4,SYTIMH       ;HIGH-ORDER TIME VALUE
011316  010567  0000000              MOV     R5,SYTIML       ;LOW-ORDER TIME VALUE
011322  000167  0000000              JMP     RDCMD           ;FINISHED
                                ;  Bad time value entered
011326  152767  000010   0000000 BADTIM: BISB    #10,INDERR      ;SAVE ERROR LEVEL = SEVERE FOR IND
011334                               FABORT  #INVTIM         ;INVALID TIME VALUE
```

```
                                        .SBTTL  RESET command
                                ;-----------------------------------------------------------------------
                                ;  The RESET command resets system usage statistics.
                                ;  Operator privilege is required.
                                ;
011344  004767  0000000        CMDRST: CALL    CKSYPV          ;Require SYSPRV privilege
                                ;  Reset data cache statistics
011350  005067  0000000              CLR     DCTRD           ;TOTAL NUMBER OF READS FROM SHARED FILES
011354  005067  0000000              CLR     DCCRD           ;NUMBER OF READS FOUND IN CACHE
011360  005067  0000000              CLR     DCTWR           ;TOTAL NUMBER OF WRITES TO SHARED FILES
011364  005067  0000000              CLR     DCCWR           ;NUMBER OF WRITES THAT UPDATE CACHE DATA
                                ;  Reset 32-bit data cells
011370  012701  011412'             MOV     #RSTVEC,R1      ;POINT TO VECTOR OF ADDRESSES TO CLEAR
011374  012102                 1$:  MOV     (R1)+,R2        ;GET ADDRESS OF A CELL TO ZERO
011376  001403                      BEQ     2$              ;BR IF HIT END OF LIST
011400  005022                      CLR     (R2)+           ;CLEAR HIGH-ORDER WORD
011402  005012                      CLR     (R2)            ;CLEAR LOW-ORDER WORD
011404  000773                      BR      1$
011406  000167  0000000        2$:  JMP     RDCMD           ;FINISHED
                                ;
                                ;  Vector of 32-bit cells to clear.
                                ;
011412  0000000              RSTVEC: .WORD   TMTOTH
011414  0000000                      .WORD   TMUSRH
011416  0000000                      .WORD   TMSWTH
011420  0000000                      .WORD   TMIOH
011422  0000000                      .WORD   TMSWPH
011424  0000000                      .WORD   TMIOWH
011426  0000000                      .WORD   TMIDLH
011430  0000000                      .WORD   CASTRO
011432  0000000                      .WORD   CASTBR
011434  0000000                      .WORD   CASCBR
011436  0000000                      .WORD   CASTWO
011440  0000000                      .WORD   CASTBW
011442  0000000                      .WORD   CASCUP
011444  000000                       .WORD   0
```

```
                                        .SBTTL  OFF command
                                ;-----------------------------------------------------------
                                ;  LOG OFF.
                                ;
011446  116701  0000000         CMDOFF: MOVB    CORUSR,R1       ;GET USER INDEX #
                                ;
                                ;  See if any qualifiers were specified with command
                                ;
011452  105067  166357                  CLRB    OFFNWF          ;Clear NOWARN flag
011456  012704  000250'                MOV     #OFFHD,R4       ;Point to parsing table
011462  004767  0000000                 CALL    OPTLST          ;Process any qualifiers
                                ;
                                ;  See if this is a primary process logging off with active subprocesses
                                ;
011466  020127  0000000                 CMP     R1,#LSTPL       ;Is this a primary process?
011472  101024                          BHI     14$             ;Br if not
011474  105767  166335                  TSTB    OFFNWF          ;Was /NOWARN qualifier specified?
011500  001021                          BNE     14$             ;Br if yes -- Don't worry about subprocesses
011502  032761  0000000 0000000         BIT     #$PRGLK,LSW5(R1);Did we exit a locked program?
011510  001015                          BNE     14$             ;Br if yes -- Don't worry about subprocesses
011512  032761  0000000 0000000         BIT     #$NOIN,LSW3(R1) ;Are we allowed to have terminal input?
011520  001011                          BNE     14$             ;Br if not
011522  032761  0000000 0000000         BIT     #$LOFCF,LSW9(R1);Have we already done a logAoff command file?
011530  001005                          BNE     14$             ;Do logoff if so
011532  004767  001142                  CALL    CKOFSP          ;See if we need to warn about subprocesses
011536  103002                          BCC     14$             ;Br if we should logoff
011540  000167  0000000                 JMP     RDCMD           ;Abort the logoff
                                ;
                                ;  Do the logoff.
                                ;
011544  052761  0000000 0000000 14$:    BIS     #$NOIN,LSW3(R1) ;DON'T ALLOW ABORT OF LOGOFF
011552  005067  0000000                 CLR     RESDEV          ;RELEASE ALL DEVICE/FILE ACCESS RESTRICTIONS
                                ;
                                ;  Close all of user's files.
                                ;
011556  004767  0000000                 CALL    PRGALL          ;PURGE ALL OF USER'S CHANNELS
011562  004767  0000000                 CALL    INDABT          ;Abort IND and nested command files
011566  004767  0000000                 CALL    ABRTCF          ;CLOSE THE CONTROL FILE
011572  052761  0000000 0000000         BIS     #$NOIN,LSW3(R1) ;CLEARED BY ABRTCF(POPCF), SET IT AGAIN
                                ;
                                ;  Determine if we need to invoke a logoff command file
                                ;
011600  005767  0000000                 TST     LOFSPC          ;Is there a logoff command file?
011604  001471                          BEQ     3$              ;Br if not
011606  004767  0000000                 CALL    PUSHCF          ;Prepare to open new command file
011612  112767  000001  0000000         MOVB    #1,SERFLG       ;Do .SERR
011620                                  .LOOKUP #XAREA,#CFCHAN,#LOFSPC ;Try to open the command file
011640  112767  000000  0000000         MOVB    #0,SERFLG       ;Do .HERR but don't affect carry flag
011646  103013                          BCC     10$             ;Br if open was ok
011650  004767  0000000         11$:    CALL    POPCF           ;Pop command file status
011654  005067  0000000                 CLR     LOFSPC          ;No logoff command file
011660                                  FERR    #EM$OLO         ;Cannot open logoff command file
011674  000435                          BR      3$
011676                          10$:    .READW  #XAREA,#CFCHAN,#CFBUF,#256.,#0 ;Read 1st block of file
011734  103745                          BCS     11$             ;Br if read error
011736  052761  0000000 0000000         BIS     #$CFOPN,LSW4(R1);Say CFCHAN is open
011744  052761  0000000 0000000         BIS     #$LOFCF,LSW9(R1);Say we are processing logoff command file
```

```
                                                BIC     #$DOOFF,LSW(R1) ;Say not doing logoff processing yet
                                                CLR     LOFSPC          ;Say logoff command file has been started
                                                JMP     RDCMD           ;Go process 1st command in command file
                                        ;
                                        ;  Enable privilege during logoff processing
                                        ;
                                        3$:     MOV     #PRIVCO,R2      ;Point to current privilege word
                                                MOV     #PRIVSO,R3      ;Point to set privileges
                                                MOV     #PVNPW,R0       ;Get # privilege words
                                        12$:    MOV     #177777,(R2)+   ;Set all privilege flags
                                                MOV     #177777,(R3)+
                                                SOB     R0,12$
                                                BIS     #LF$WRT,LOGFLG  ;ENABLE WRITES TO LOG FILE
                                        ;
                                        ;  Dismount all devices that were mounted by this job.
                                        ;
                                                CALL    DMTALL          ;Dismount all mounted devices
                                        ;
                                        ;  Print usage statistics for line
                                        ;
                                                BIT     #$VNOTT,LSW(R1) ;IS THIS JOB CONNECTED TO A TERMINAL?
                                                BNE     7$              ;BR IF NOT
                                                MOV     LNPRIM(R1),R0   ;GET PRIMARY LINE #
                                                BIC     #$CTRLS,LSW3(R0);MAKE SURE OUTPUT IS ENABLED
                                                CMP     R1,#LSTDL       ;Is this a subprocess?
                                                BLOS    13$             ;Br if not
                                                TST     LWINDO(R0)      ;Does primary process have windowing on?
                                                BNE     7$              ;Br if yes -- Don't display logoff time
                                        13$:    .PRINT  #CRLF
                                                TSTB    STPFLG          ;IS SYSTEM BEING SHUT DOWN?
                                                BEQ     5$              ;BR IF NOT
                                                .PRINT  #SHTMSG         ;PRINT SHUT-DOWN MESSAGE
                                        5$:     CALL    PRTTIM          ;PRINT USAGE INFO
                                                .TTYOUT #LF             ;PUT OUT EXTRA LF
                                                BR      TIMUP
                                        ;  IF PRTTIM DIDN'T DO CPU CALCULATION WE MUST DO IT HERE.
                                        7$:     MOV     LCPUHI(R1),R4   ;GET HIGH-ORDER CPU TIME (CLOCK TICKS)
                                                MOV     LCPULO(R1),R5   ;GET LOW-ORDER CPU TIME (CLOCK TICKS)
                                                MOV     TK1SEC,R3       ;GET # CLOCK TICKS PER SECOND
                                                CALL    DIVIDE          ;CONVERT TIME VALUE TO # SECONDS
                                                MOV     #10.,R0         ;CONVERT TIME VALUE TO 0.1 SECOND
                                                CALL    MUL32           ;UNITS BY MULTIPLYING BY 10
                                                MOV     R4,CPUAH        ;SAVE HIGH-ORDER PART (0.1 SEC)
                                                MOV     R5,CPUAL        ;SAVE LOW-ORDER PART (0.1 SEC)
                                        ;
                                        ;  If user logged on, then accumulate his connect time
                                        ;
                                        TIMUP:  MOV     LPROJ(R1),R5    ;DO WE HAVE A PPN FOR USER?
                                                BEQ     NOTIM           ;BR IF NOT -- DIDN'T LOG ON
                                        ;  DO A DEASSIGN
                                                MOV     #ASNTBL,R2      ;CLEAR ALL OF HIS ASSIGNS
                                        1$:     CLR     (R2)+
                                                CMP     R2,#ASNEND
                                                BLO     1$
                                        ;  Try to open accounting file
                                                .LOOKUP #XAREA,#1,#AUTHFN
                                                BCC     2$              ;BR IF OPEN OK
```

```
                                        .PRINT  #COAF           ;PRINT WARNING MESSAGE
                                        BR      NOTIM
                                ;  Search file for user's ppn entry
                                2$:     CLR     R2              ;SET BLOCK # TO 0
                                        MOV     LPROG(R1),R3    ;GET PROGRAMMER #
                                6$:     .READW  #XAREA,#1,#BLKO,#256.,R2
                                        BCS     3$              ;BR IF END OF FILE HIT
                                        CALL    CVTBUF          ;COMPLEMENT CONTENTS OF BUFFER
                                        MOV     #BLKO,R4        ;SEARCH FOR RECORD IN BLOCK
                                        MOV     #ARNRPB,R0      ;# RECORDS PER BLOCK
                                5$:     CMP     R5,AR$PRJ(R4)   ;DO PROJECT #'S MATCH?
                                        BNE     8$              ;BR IF NOT
                                        CMP     R3,AR$PRG(R4)   ;HOW ABOUT PROGRAMMER NUMBERS
                                        BEQ     4$              ;BR IF WE FOUND THE RECORD
                                8$:     ADD     #AR$$SZ,R4      ;KEEP LOOKING IN BLOCK
                                        DEC     R0              ;ANY MORE RECORDS IN THIS BLOCK?
                                        BNE     5$              ;BR IF YES
                                        INC     R2              ;GO READ NEXT BLOCK
                                        BR      6$
                                ;  Found record.  Add in new connect time
                                4$:     MOV     MINTIM,R0       ;GET CURRENT TIME
                                        SUB     LCONTM(R1),R0   ;CALCULATE CONNECT TIME
                                        INC     R0
                                7$:     ADD     R0,AR$CON(R4)   ;ACCUMULATE CONNECT TIME
                                ;  COUNT NUMBER OF SESSIONS
                                        INC     AR$CNT(R4)      ;INC SESSION COUNTER
                                ;  ACCUMULATE CPU TIME
                                        ADD     CPUAL,AR$CPL(R4)        ;ADD IN LOW-ORDER CPU TIME
                                        ADC     AR$CPH(R4)      ;ADD CARRY
                                        ADD     CPUAH,AR$CPH(R4)        ;ADD IN HIGH-ORDER PART
                                ;  WRITE UPDATED ACCOUNTING RECORD BACK TO FILE
                                        CALL    CVTBUF          ;COMPLEMENT CONTENTS OF BUFFER
                                        .WRITW  #XAREA,#1,#BLKO,#256.,R2
                                3$:     .CLOSE  #1
                                ;
                                ;  Close job's log file
                                ;
                                NOTIM:  CALL    LOGCLS          ;CLOSE LOG FILE
                                ;
                                ;  Broadcast message to any monitoring jobs telling them we are logging off
                                ;
                                        MOV     #GENMON,R0      ;Point to emt argument block
                                        MOV     #JS$OFF,2(R0)   ;Set status code
                                        EMT     375             ;Broadcast status message
                                ;
                                ;  Enter TSX to finish log off.
                                ;
                                        TSTB    STPFLG          ;ARE WE DOING A SYSTEM SHUTDOWN?
                                        BEQ     1$              ;BR IF NOT
                                        BGT     15$             ;Br if $STOP or BOOT specified
                                ;
                                ;  Shutdown was specified.
                                ;
                                        BIS     #$NOABT,LSW2(R1);Don't allow job abort
                                        CMPB    R1,#LSTPL       ;Is this a detached job?
                                        BLO     10$             ;Br if not detached
                                        CMPB    R1,#LSTDL       ;In the detached range?
```

```
     172 012536  101435                          BLOS    15$             ;Yes, job is detached
     173 012540  126727  0000000 000001  10$:    CMPB    PVON,#1         ;Are we the last primary/virtual job?
     174 012546  003031                          BGT     15$             ;Br if more jobs exist
     175                                ;
     176                                ;  Kill all jobs (only detached jobs should be active).
     177                                ;
     178 012550  012702  0000000                 MOV     #LSTSL,R2       ;GET # OF LAST LINE
     179 012554  120267  0000000         11$:    CMPB    R2,CORUSR       ;IS THIS OUR LINE?
     180 012560  001405                          BEQ     12$             ;BR IF YES
     181 012562  012700  0000000                 MOV     #KILEMT,R0      ;POINT TO KILL EMT ARG BLOCK
     182 012566  010260  000004                  MOV     R2,4(R0)        ;SET # OF LINE TO KILL
     183 012572  104375                          EMT     375             ;KILL THE LINE
     184 012574  162702  000002          12$:    SUB     #2,R2           ;DO NEXT LINE
     185 012600  001365                          BNE     11$
     186                                ;
     187                                ;  Determine if the spooler is still active.
     188                                ;
     189 012602  004767  0000000         13$:    CALL    SPLACT          ;Is the spooler still active?
     190 012606  103011                          BCC     15$             ;Br if spooler is not active.
     191 012610                                  .TWAIT  #XAREA,#TIMSPL  ;Wait 2 seconds -- check again
     192 012630  000764                          BR      13$
     193                                ;
     194                                ;  $STOP or BOOT issued.
     195                                ;
     196 012632  042761  0000000 0000000 15$:    BIC     #$NOABT,LSW9(R1);Allow job abort (necessary for log off)
     197 012640  126727  0000000 000001         CMPB    TOTON,#1        ;Are we the last job on system?
     198 012646  003002                          BGT     1$              ;Br if more jobs exist
     199 012650  000167  0000000                 JMP     DOSTOP          ;GO STOP THE SYSTEM
     200 012654  005061  0000000          1$:    CLR     LPROJ(R1)       ;CLEAR PROJECT/PROGRAMMER #
     201 012660  005061  0000000                 CLR     LPROG(R1)
     202 012664  012700  0000000                 MOV     #OFFEMT,R0      ;POINT TO LOG-OFF EMT ARG BLOCK
     203 012670  104375                          EMT     375             ;LOG OFF THIS JOB
```

```
                                ;-------------------------------------------------------------------------------
                                ;  Subroutine called if the /NOWARN qualifier is specified with the OFF
                                ;  command.
                                ;
012672  105267  165137          OFFNWN: INCB    OFFNWF          ;Set NOWARN flag
012676  000207                          RETURN
                                ;
                                ;-------------------------------------------------------------------------------
                                ;  Subroutine called to determine if primary process has any active
                                ;  subprocesses and print warning message if so.
                                ;
                                ;  Inputs:
                                ;   R1 = Process index number
                                ;
                                ;  Outputs:
                                ;   C-flag cleared ==> Proceed with logoff.
                                ;   C-flag set     ==> Abort the logoff.
                                ;
012700  010246                  CKOFSP: MOV     R2,-(SP)
012702  010546                          MOV     R5,-(SP)
                                ;
                                ;  See if this process has any active subprocesses
                                ;
012704  005005                          CLR     R5              ;Count active subprocesses in R5
012706  016102  000000G                 MOV     LSECPT(R1),R2   ;Point to table of subprocess #'s
012712  012700  000000G                 MOV     #MAXSEC,R0      ;Get max # subprocesses
012716  001454                          BEQ     7$              ;Br if none to check
012720  105722                  1$:     TSTB    (R2)+           ;Is this subprocess active?
012722  001401                          BEQ     4$              ;Br if not
012724  005205                          INC     R5              ;Count # active subprocesses
012726  077004                  4$:     SOB     R0,1$           ;Loop to check all
012730  005705                          TST     R5              ;Any active subprocesses?
012732  001446                          BEQ     7$              ;Br if not
                                ;
                                ;  There are active subprocesses.
                                ;  Print warning message.
                                ;
012734  020527  000001                  CMP     R5,#1           ;One active subprocess?
012740  003004                          BGT     2$              ;Br if more than one
012742                                  .PRINT  #TM$SA1         ;You have 1 active subprocess
012750  000410                          BR      5$
012752                          2$:     .PRINT  #TM$SA3         ;You have
012760  004767  000000G                 CALL    PRTDEC          ;Print # subprocesses
012764                                  .PRINT  #TM$SA4         ;active subprocesses
                                ;
                                ;  Read response
                                ;
012772                          5$:     .GTLIN  #BLK0,#TM$SA2   ;Print prompt and read response
                                ;
                                ;  If first non-blank letter of response is "Y" then logoff,
                                ;  otherwise abort the logoff.
                                ;
013012  012702  000000G                 MOV     #BLK0,R2        ;Point to response buffer
013016  112200                  3$:     MOVB    (R2)+,R0        ;Get next char of response
013020  001411                          BEQ     8$              ;Br if hit end of line
013022  120027  000040                  CMPB    R0,#40          ;Ignore spaces
013026  001773                          BEQ     3$
```

```
      58 013030  120027  000131                    CMPB    R0,#'Y          ;Is response Yes?
      59 013034  001405                            BEQ     7$              ;Br if yes
      60 013036  120027  000171                    CMPB    R0,#'y          ;Allow lower-case too
      61 013042  001402                            BEQ     7$
      62                                ;
      63                                ;  Abort the logoff
      64                                ;
      65 013044  000261                 8$:        SEC                     ;Signal abort on return
      66 013046  000401                            BR      9$
      67                                ;
      68                                ;  Allow the logoff
      69                                ;
      70 013050  000241                 7$:        CLC                     ;Signal to logoff
      71                                ;
      72                                ;  Finished
      73                                ;
      74 013052  012605                 9$:        MOV     (SP)+,R5
      75 013054  012602                            MOV     (SP)+,R2
      76 013056  000207                            RETURN
```

```
                                        ;-------------------------------------------------------------------------------
                                        ;  Complement the contents of BLKO buffer.
                                        ;  We do this to mildly encript the authorization file.
                                        ;
013060  010146                  CVTBUF: MOV     R1,-(SP)
013062  010246                          MOV     R2,-(SP)
013064  012701  0000000                 MOV     #BLKO,R1        ;POINT TO BUFFER
013070  012702  000400                  MOV     #256.,R2        ;GET # WORDS TO COMPLEMENT
013074  005121                  1$:     COM     (R1)+           ;COMPLEMENT CONTENTS OF BUFFER
013076  077202                          SOB     R2,1$
013100  012602                          MOV     (SP)+,R2
013102  012601                          MOV     (SP)+,R1
013104  000207                          RETURN
```

```
                                        .SBTTL  KILL command
                                ;-------------------------------------------------------------------------
                                ;  The KILL command is used to abort the specified job.
                                ;
013106  012705  0000000         CMDKIL: MOV     #KILEMT,R5      ;Point to Kill EMT arg block
013112  000405                          BR      JOBCOM
                                
                                        .SBTTL  SUSPEND command
                                ;-------------------------------------------------------------------------
                                ;  Suspend the execution of a specified job
                                ;
013114  012705  0000000         CMDSPN: MOV     #SJEMT,R5       ;Point to suspend EMT arg block
013120  000402                          BR      JOBCOM
                                
                                        .SBTTL  RESUME command
                                ;-------------------------------------------------------------------------
                                ;  Resume the execution of a suspended job
                                ;
013122  012705  0000000         CMDRSM: MOV     #RJEMT,R5       ;Point to resume EMT arg block
                                ;
                                ;  Accrue the job number
                                ;
013126  004767  0000000         JOBCOM: CALL    CVTTAB          ;convert tab and FF chars to spaces
013132  004767  0000000                 CALL    ACRDEC          ;accrue decimal value
013136  006301                          ASL     R1              ;Convert # to job index number
013140  001403                          BEQ     1$              ;Br if invalid line number
013142  020127  0000000                 CMP     R1,#LSTSL       ;Is this a valid line number
013146  101404                          BLOS    2$              ;Br if ok
013150                          1$:     FABORT  #BDLIN          ;Invalid line number
                                ;
                                ;  See if we are privileged to affect this job
                                ;
013160  010102                  2$:     MOV     R1,R2           ;Get job # to R2 for CKACOJ
013162  004767  0000000                 CALL    CKACOJ          ;Can we access this job
013166  103404                          BCS     9$              ;Br if not
                                ;
                                ;  Execute EMT to affect the job
                                ;
013170  010500                          MOV     R5,R0           ;Point to EMT arg list
013172  010160  000004                  MOV     R1,4(R0)        ;Set job number
013176  104375                          EMT     375             ;Perform the function
                                ;
                                ;  Finished
                                ;
013200  000167  0000000         9$:     JMP     RDCMD
```

```
                                        .SBTTL  BOOT and $STOP commands
                                ;-------------------------------------------------------------------------------
                                ;  Reboot RT-11.
                                ;
013204  004767  0000000         CMDBOT: CALL    CKPRIV          ;USER MUST HAVE OPERATOR COMMAND PRIVILEGE
013210  004767  0000000                 CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACES
                                ;
                                ;  See if a boot device was specified.
                                ;
013214  005067  0000000                 CLR     BOTDEV          ;ASSUME NO BOOT DEVICE WILL BE SPECIFIED
013220  105713                          TSTB    (R3)            ;WAS A BOOT DEVICE SPECIFIED?
013222  001405                          BEQ     7$              ;BR IF NOT -- USE SYSTEM DEVICE
                                ;  Accrue device name.
013224  004767  0000000                 CALL    GTRD50          ;ACCRUE RAD50 DEVICE NAME
013230  016767  0000000 0000000         MOV     R50BUF,BOTDEV   ;SAVE RAD50 NAME OF BOOT DEVICE
                                ;
                                ;  Make sure system is idle.
                                ;
                                ;  See if any other users are logged on.
013236  005005                  7$:     CLR     R5              ;IDLE FLAG
013240  126727  0000000 000001        CMPB    NUMON,#1        ;ARE WE ONLY JOB LOGGED ON?
013246  003404                          BLE     1$              ;BR IF YES
013250                                  .PRINT  #OTHRON         ;OTHER USERS LOGGED ON
013256  005205                          INC     R5              ;REMEMBER NOT IDLE
                                ;  See if the spooler is idle.
013260  004767  0000000         1$:     CALL    SPLACT          ;SEE IF SPOOLER IS ACTIVE
013264  103004                          BCC     2$              ;BR IF SPOOLER IS IDLE
013266                                  .PRINT  #SPLPND         ;THERE ARE PENDING SPOOL FILES
013274  005205                          INC     R5              ;REMEMBER SYSTEM IS NOT IDLE
                                ;  See if we need to ask for confirmation.
013276  005705                  2$:     TST     R5              ;WAS SYSTEM IDLE?
013300  001420                          BEQ     10$             ;BR IF YES
013302                                  .PRINT  #STPASK         ;ASK FOR CONFIRMATION
013310                          5$:     .TTYIN                  ;ACCEPT INPUT LINE
013314  120027  000131                  CMPB    R0,#'Y          ;DID HE SAY YES?
013320  001410                          BEQ     10$             ;BR IF YES
013322  120027  000171                  CMPB    R0,#171         ;LOWER-CASE 'Y'
013326  001405                          BEQ     10$
013330  120027  000012                  CMPB    R0,#LF          ;EAT REST OF LINE
013334  001365                          BNE     5$
013336  000167  0000000                 JMP     RDCMD           ;GO GET NEXT COMMAND
                                ;
                                ;  Force logoff of all users.
                                ;
013342  112767  000001  0000000 10$:     MOVB    #1,STPFLG       ;SET FLAG SAYING SYSTEM IS BEING STOPPED
013350  012702  0000000                 MOV     #LSTSL,R2       ;GET # OF LAST LINE
013354  120267  0000000         11$:    CMPB    R2,CORUSR       ;IS THIS OUR LINE?
013360  001405                          BEQ     12$             ;BR IF YES
013362  012700  0000000                 MOV     #KILEMT,R0      ;POINT TO KILL EMT ARG BLOCK
013366  010260  000004                  MOV     R2,4(R0)        ;SET # OF LINE TO KILL
013372  104375                          EMT     375             ;KILL THE LINE
013374  162702  000002          12$:    SUB     #2,R2           ;DO NEXT LINE
013400  001365                          BNE     11$
                                ;
                                ;  Now enter logoff processing for our line.
                                ;  It will jump to DOSTOP after it finishes.
                                ;
```

```
      58 013402  000167  176040                    JMP     CMDOFF          ;LOG OFF OUR LINE
```

```
      1                                          .SBTTL  $SHUTDOWN command
      2                                ;-----------------------------------------------------------------------
      3                                ;  Do a gentle shutdown of system.  Don't force jobs off but don't let
      4                                ;  any log on either.
      5                                ;
      6 013406  004767  0000000        CMDSHT: CALL    CKPRIV          ;USER MUST HAVE OPERATOR PRIV
      7 013412  112767  177777  0000000       MOVB    #-1,STPFLG      ;SAY SYSTEM SHUTDOWN TAKING PLACE
      8 013420  005067  0000000               CLR     BOTDEV          ;REBOOT FROM SYSTEM DISK
      9 013424  000167  0000000               JMP     RDCMD
     10            000001                    .END
Errors detected:  0

*** Assembler statistics


Work  file  reads:  0
Work  file writes:  0
Size of work file:  12080 Words  ( 48 Pages)
Size of core pool:  17920 Words  ( 70 Pages)
Operating  system:  RT-11

Elapsed time:  00:01:40.48
DK:TSKM2A,LP:TSKM2A=DK:TSKM2A.MAC/C/N:SYM
```

Cross reference table (CREF V05.04)

```
$ISTLG    1-68
$CARUP    1-73
$CCLRN    1-74
$CFABT    1-91        7-17        7-34
$CFALL    1-98
$CFCCL    1-98
$CFDCC    1-98
$CFOPN    1-104      42-56
$CFSOT    1-96
$CHACT    1-66
$CLTST    1-83
$CTRLC    1-89
$CTRLD    1-133
$CTRLO    1-49
$CTRLS    1-79       42-81
$DBKMN    1-72
$DEAD     1-136
$DEBUG    1-134
$DEFER    1-110
$DETCH    1-77       33-36       33-124
$DIBOL    1-68
$DILUP    1-93       33-97       33-122
$DISCN    1-78
$DOOFF    1-100      42-58
$DUPRN    1-95
$ECHO     1-97
$EMTTR    1-82
$FORM     1-96
$FORMO    1-98
$HARD     1-136
$HITTY    1-67
$INCOR    1-113
$INDDF    1-135
$INDRN    1-135
$INIT     1-136
$INKMN    1-89       33-103      33-132
$KED      1-113
$KINIT    1-57
$LC       1-97
$LOFCF    1-181      42-23       42-57
$NOABT    1-184      42-168      42-196
$NOIN     1-67       42-21       42-31       42-39
$NOINT    1-182
$NOWIN    1-38
$NOWTT    1-67
$PAGE     1-97
$PHONE    1-136
$PRGLK    1-75       42-17
$QTSET    1-118
$QUIET    1-111       7-21
$RNIOP    1-183
$RNMLK    1-183
$SCCA     1-39        7-15
$SCOPE    1-97
$SETRN    1-97
$SGALL    1-110
```

Cross reference table (CREF V05.04)

```
$SGHIO    1-109
$SGIIO    1-109
$SGQO     1-90
$SGQ1     1-109
$SGQ1A    1-109
$SGQ1B    1-109
$SGQ1C    1-109
$SGQ2     1-109
$SGQ3     1-90
$SLLET    1-180
$SLON     1-180
$SLTTY    1-180
$SNWTT    1-134
$SPLJB    1-79
$STSNG    1-66
$SUCF     1-99      35-5
$TAB      1-96
$TAPE     1-132
$TECO     1-118
$TRNSP    1-42
$TTGAG    1-104     33-101    33-128
$UCLCF    1-105
$UCLCL    1-107
$UCLCM    1-107
$UCLRN    1-106
$UKMON    1-73
$VNOTT    1-48      42-78
$WILD     1-119
. . . V1   7-23      7-27      7-50      8-63      11-71     12-53     14-16     14-24     14-37     22-50     23-91     25-54
          28-47     30-33     31-11     31-18     31-18     31-18     31-20     31-21     31-21     31-27     31-27     31-32
          31-32     31-34     31-34     31-38     31-38     31-39     31-39     32-6      32-10     32-12     32-16     32-20
          32-22     32-27     32-33     32-35     32-39     32-41     32-42     33-99     33-107    37-39     37-41     37-49
          38-11     38-11     38-11     38-13     38-13     38-26     38-28     42-47     42-47     42-47     42-54     42-54
          42-86     42-89     42-91     42-113    42-113    42-113    42-115    42-120    42-120    42-147    42-147    42-148
          42-148    42-191    43-40     43-42     43-44     43-48     46-23     46-28     46-33
. . . V2   23-91     23-91     23-91#    23-91#    31-18     31-18     31-18#    31-18#    31-21     31-21#    31-27     31-27
          31-27#    31-27#    31-32     31-32     31-32#    31-32#    31-34     31-34     31-34#    31-34#    31-38     31-38#
          31-39     31-39#    38-11     38-11     38-11#    38-11#    38-13     38-13#    42-47     42-47     42-47#    42-47#
          42-54     42-54     42-54#    42-54#    42-113    42-113    42-113#   42-113#   42-120    42-120    42-120#   42-120#
          42-147    42-147    42-147#   42-147#   42-148    42-148#
ABRTCF    1-143     42-30
ACBDFS    35-53     35-67     35-102#
ACBDSW    35-87     35-119#
ACRDEC    1-151     28-21     30-19     31-84     34-52     37-21     39-14     39-39     40-14     40-22     40-33     45-24
ACRFIL    1-150     18-32     23-34     38-9
ACRFN     1-143
ACROCT    1-153     28-11     28-15
AD$$SZ    1-52
AD$DVU    1-52
AD$JOB    1-52
AF$BYA    1-65      5-6
AF$CCA    1-39      5-19
AF$DBG    1-43      5-8
AF$DUP    1-40      5-7
AF$HIE    1-182     5-9
AF$IND    1-40      5-10
```

Cross reference table (CREF V05.04)

```
AF$IOP    1-183     5-11      5-12
AF$MEM    1-65      5-14
AF$NOI    1-182     5-15
AF$NOW    1-182     5-16
AF$NPW    1-39      5-17
AF$PLK    1-43      5-13
AF$SCA    1-68      5-21
AF$SET    1-40      5-20
AF$TPO    1-42      5-22
AF$UCL    1-40      5-23
AFCF      1-39
ALC1DV    4-6#      11-11*    11-43     11-45*    11-86     12-5*     12-31*    12-61
ALCDEV    1-28      11-39*    12-43*    12-63*    22-41*    26-38*    26-68
ALDBLK    1-166     31-17*    31-18
ALDEMT    1-76      11-49
ALDEX     1-165      1-166    31-11     31-11
ALFN      1-177     31-27
AMBOPT    1-151      1-164    29-65
AR$$SZ    1-180     42-129
AR$CNT    1-179     42-140*
AR$CON    1-179     42-138*
AR$CPH    1-179     42-143*   42-144*
AR$CPL    1-179     42-142*
AR$DMY    1-180
AR$PRG    1-179     42-127
AR$PRJ    1-179     42-125
AR$UNM    1-179
ARNRPB    1-180     42-124
ASCIS     8-53       8-62#
ASCSOK    8-65       8-81#
ASDELM    8-17       9-7#
ASDEX     1-148      8-63
ASFID     8-36       8-57#
ASKLNM    1-147      8-8*      8-52*     8-85
ASNEND    1-95      10-14     10-28     42-110
ASNHD1    1-160
ASNHD2    1-160
ASNOVF    1-148      1-173     8-97
ASNSRC    1-122      8-82      8-90      8-94      25-31     29-21     35-45
ASNTBL    1-93      10-12     10-22     42-108
ASSMPL    8-42#
AT$$SZ    1-94      10-27
AT$DEV    1-94       8-84     8-105*    10-26*    25-33     29-23     35-47
AT$EXT    1-94       8-108*
AT$FIL    1-94       8-106*   8-107*
AT$LOG    1-94       8-102*   10-23     10-25*
AT$SIZ    1-94       8-103*
AUTHFN    1-170     42-113
BADCMD    1-147      8-15     29-12
BADDAT   39-16      39-18     39-22     39-33     39-37     39-41     39-43     39-49#
BADPMT    1-154
BADPRI    1-154
BADSAV    1-147
BADTIM   40-16      40-20     40-24     40-35     40-49#
BASMAP    1-127
BDFNAM    1-150     18-99     23-36     24-17     38-37
```

Cross reference table (CREF V05.04)

```
BDLGOP   1-155
BDLIN    1-168    33-34    37-25    45-29
BELL     2-7#     33-49
BLANK    2-6#     33-65
BLKO     1-146    31-32    31-34    42-120   42-123   42-147   43-48    43-53    44-7
BLKWDS   2-10#
BOTDEV   1-88     46-10*   46-15*   47-8*
BOTUNI   1-88
C.CSW    1-101    27-27
C.DEVQ   1-101    27-31
C.SBLK   1-101
CALUCL   1-159
CASCBR   1-84     41-37
CASCUP   1-84     41-35
CASTBR   1-84     41-31
CASTBW   1-84     41-34
CASTRO   1-85     41-30
CASTWO   1-85     41-33
CCLNAM   1-174
CCLSAV   1-81     27-59
CCSPRV   1-181
CD$$SZ   1-102    26-115
CD$$UB   1-102
CD$BAS   1-102    26-76
CD$DVU   1-102    26-94
CD$JOB   1-102
CD$NAM   1-102
CD$TOP   1-118
CDBUF    1-47     26-94    26-96
CDGET    1-47     26-93
CF$IND   1-137
CF$QUT   1-137
CFABLV   1-138
CFBLK    1-111
CFBUF    1-81     42-54
CFCHAN   1-110    42-47    42-47    42-54    42-54
CFEND    1-81
CFHOLD   1-117
CFIND    1-100
CFLFL4   1-105
CFNEST   1-103
CFPNT    1-111     7-8
CFSEND   1-104
CFSP     1-104
CFSPND   1-116     7-38      7-35     7-37*
CFSQEZ   1-42
CFSTK    1-57
CFSTOP   1-41      7-10
CFSTRT   1-41      7-36
CFSTS    1-137
CHAIN    1-89
CHKALC   1-66     26-69
CHKDEV   1-174    13-30    18-53    26-75    27-18
CHKDLM   1-164    29-49    35-63    35-91
CHKMNT   1-149    25-50    26-48
CHKMTX   1-149    25-52    26-54    26-106
```

Cross reference table (CREF V05.04)

```
CINDAT    1-126
CINFLG    1-48
CKACOJ    1-44      45-34
CKOFSP   42-25      43-19#
CKPRIV    1-151     31-5      31-44     31-50     31-78     31-90     32-48     32-54     32-61     32-70     33-6      39-12
         40-12      46-5      47-6
CKSYDV   26-42      27-11#
CKSYPV    1-44      16-4      17-4      41-6
CLDEVX    1-124
CLOTIR    1-53
CLRPRV    1-46      16-6
CLSFSP    1-85
CLTOTL    1-85
CMDACC    1-32      35-5#     35-114
CMDALC    1-31      11-7#
CMDASN    1-30       8-7#
CMDBOT    1-33      46-5#
CMDBUF    1-142      1-159     7-23
CMDCCL    1-174     26-127
CMDDAT    1-29      39-5#
CMDDET    1-32      37-5#
CMDDLC    1-31      12-5#
CMDDMT    1-30      25-5#
CMDDSN    1-27      10-5#
CMDDSP    1-30       7-44#    7-45
CMDFMT    1-29      26-10#
CMDFRM    1-27      32-79#
CMDHD     1-26
CMDINI    1-32      26-8#
CMDINS    1-28      15-5#
CMDKIL    1-31      45-5#
CMDMNT    1-30      22-8#
CMDMON    1-30      28-7#
CMDOFF    1-26      42-5#     46-58
CMDPAU    1-32       7-8#
CMDREM    1-31      13-8#
CMDRSM    1-28      45-19#
CMDRST    1-29      41-6#
CMDRSY    1-29
CMDSHT    1-33      47-6#
CMDSND    1-29      33-21#
CMDSPN    1-28      45-12#
CMDSPO    1-31      29-8#
CMDSQZ    1-32      26-9#
CMDTIM    1-30      40-5#
CMDYEL    1-28      33-6#
COAD      1-165      1-166    31-20
COAF      1-178     42-115
COAL      1-165      1-166    31-8
COLOO     1-159
CONFIG    1-99      40-40
CORUSR    1-49      33-50     33-120    34-40     35-113    42-5      42-179    46-47
CPUAH     1-159     42-100*   42-144
CPUAL     1-159     42-101*   42-142
CR        2-5#      33-47     33-84
CRLF      1-154     11-71     12-53     22-50     28-47     32-35     38-28     42-86
```

Cross reference table (CREF V05.04)

```
CSABT    30-32#     30-34
CSCD     33-33      33-36#
CSEXIT   33-23      33-35#
CSGTOP   29-37      29-48#     29-50
CSHALC    1-88
CSHDEV    1-93      26-71
CSHDVN    1-93      26-116
CSHHD     1-141
CSHMSG    1-181
CSHSIZ    1-128
CSIERR    8-67#
CSIMS1    1-153     28-55
CSIMS2    1-151     35-102
CSIMS4    1-173     23-82
CSSCAN   33-29      33-43#
CTRLTT    1-87      33-14      34-35
CURMTX    1-164
CURPRM    1-115
CVDVNM    1-161
CVTBUF   42-122     42-146     44-5#
CVTDEV   35-52      36-13#
CVTTAB    1-142      8-7       10-5       11-12      12-12      13-8       15-5       22-8       25-5       26-20      28-9       29-8
         32-79      35-10      37-5       39-5       40-5       45-23      46-6
CVTUC     1-141     12-13      15-6       29-9       32-80
CW$SOH    1-99      40-40
CXTBAS    1-92
CXTPAG    1-70
CXTWDS    1-92
DAJMP    10-16#     10-30
DATTIM    1-27
DCCRD     1-122     41-9*
DCCWR     1-122     41-11*
DCTRD     1-122     41-8*
DCTWR     1-122     41-10*
DEADEV    1-149     25-77
DEAS1    10-7       10-20#
DELEMT    4-24#     30-20*     30-21
DELSPC    1-163     28-10
DETARG    1-178     37-28      37-30*     38-17
DETCHK    5-48      37-36#
DETGO    37-13      38-6#
DETHD     1-178      5-47#     37-16
DETIDL   37-38      37-41#     37-48
DETJMP   37-40      37-42#     37-50      38-29
DETKIL    5-49      37-46#
DETTXT    1-157
DEVHD1    1-160
DEVIDL    1-165      1-167      1-166      30-33      32-12
DEVUNT    1-152
DFJMEM    1-57
DIVIDE    1-158     42-77
DIVSOR    1-171
DJABMS    1-169     37-49
DKSAV     1-142
DLCEMT    1-28      12-44      12-64
DLMSG     1-169     38-26
```

Cross reference table (CREF V05.04)

```
DLTXT     1-156
DMTALL    1-169     42-74
DMTARG    1-148     23-26     25-59
DMTLGQ    5-38       5-40     25-87#
DMTNLG    4-17#     25-9*     25-47     25-86*
DMTNLQ    5-39       5-41     25-86#
DMTQHD    5-37#     25-16
DNTSUB    1-174     26-111
DMYDEV    1-48      29-34
DOASGN    1-73      11-87     23-106
DOMNT    23-111     24-8#
DOSTOP    1-170     42-179
DZTXT     1-176
EDIT      1-68
EDTFIL    1-162
EM$ALC    1-41      11-22     12-8
EM$ATF    1-63      11-66
EM$CSE    1-42      13-46
EM$DAA    1-66      11-59     12-51     22-48
EM$DET    1-56      37-8
EM$DIU    1-66      11-69
EM$DNR    1-41      26-33
EM$FNI    1-46      17-10
EM$FOE    1-97      38-41
EM$IAD    1-63       1-63     11-63     12-57     18-58
EM$ITF    1-46      19-19
EM$LDI    1-43      18-57
EM$MPV    1-56
EM$NAD    1-45
EM$NAL    1-43
EM$NFW    1-41      26-16
EM$NID    1-64      30-24
EM$NLD    1-63      23-8
EM$NLN    1-43      34-19
EM$NPD    1-63
EM$NSF    1-64      35-9
EM$NUC    1-39
EM$NUK    1-74
EM$OLO    1-64      42-52
EM$SND    1-41      33-45
EM$SSY    1-64      26-44
EM$UAR    1-61      14-30
EM$UER    1-61      14-42
ERRLOC    1-55       8-67     11-56     12-48     22-45     24-13     28-40
ERRSEV    1-119
FF        2-9#
FILNAM    1-143      1-144    16-22     18-34     18-38*    18-43     18-52     18-69     18-73     23-40     23-45     23-54
         23-63      38-11
FIXPRV    1-43
FKILL     1-143      8-70      8-97     11-22     11-63     11-66     12-8      12-26     12-57     13-46     15-27     17-10
          18-57      18-58     18-99     19-19     22-21     22-28     23-8      23-15     23-36     23-50     23-82     25-24
          26-16      26-33     26-44     26-56     28-30     28-42     28-55     29-43     29-65     29-66     30-24     31-8
          33-16      33-45     34-19     34-37     35-9      35-17     35-119    37-8      37-19     37-25     38-33     38-37
          38-41      39-50     40-50     45-29
FORCEO    1-40      18-47     29-29
FPRINT    1-143     11-59     11-69     12-51     14-30     14-42     22-48     24-17     28-44     42-52
```

Cross reference table (CREF V05.04)

```
FSTDL     1-77      37-38
FSTIOL    1-50       1-70
GAGMSG    1-168     33-107
GENMON    1-65      42-154
GENTOP    1-88
GRT1      1-129
GTRD50    1-148     10-20     11-30     11-84     12-33     13-18     15-11     22-29     23-103    25-28     26-34     29-16
         35-35      35-65     35-76     39-24     46-14
HANBSY    1-153
HANCHN    1-50
HANIDX    1-152
HIMAP     1-127
HIPRI     1-154
HNBUF     1-152
HUPARG    1-173
IATTRB    4-13#     16-5*     16-35     21-31*
II$$SZ    1-182     18-78     19-13
II$FLG    1-183     16-30*
II$NAM    1-183     16-21     17-14*    18-66     18-71     18-73     19-11
II$NPV    1-45       1-183    16-39
II$PRV    1-45       1-183    16-37
IIBUF     1-47      16-21     16-35*    16-37     16-39     17-14*    18-66     18-71     18-73     19-11
ILLCMD    1-149     15-27     22-28     25-24
ILSW2     1-71
IN$ACT    1-135
IN$CMD    1-135
IN$CNT    1-135
INDABT    1-143     42-37
INDACT    1-145
INDERR    1-91      39-49*    40-49*
INDSAV    1-135     27-60
INDSTA    1-91
INDTSV    1-38      27-61
INFOMT    1-149     25-54
INGADR    1-47      18-63#    19-8*
INGEMT    1-47      18-64     19-9
INPADR    1-46      20-7*
INPEMT    1-46      20-8
INSADD   15-14      16-4#
INSATR    5-6        5-7       5-8       5-9       5-10      5-11      5-12      5-13      5-14      5-15      5-16      5-17
          5-19       5-20      5-21      5-22      5-23     21-31#
INSDEL   15-16      15-18     17-4#
INSFRE   16-17      19-7#
INSHD     5-5#      21-14
INSNAM   16-11      17-8      18-13#
INSOPT   16-7       16-30     21-10#
INSPUT   16-48      17-10     20-7#
INSSRC    1-40
INSTBL    1-182     18-62     19-7
INSTBN    1-182     18-79     19-14
INVDAT    1-178     39-50
INVDEV    1-177     29-43
INVEC     1-136
INVLDM    1-173     23-50
INVLDN    1-150     23-15
INVOPT    1-143      1-149     1-164     12-26     22-21     28-30     29-66     35-119    37-19
```

Cross reference table (CREF V05.04)

```
INVTIM    1-169     40-50
IOABFL    1-49
ITRMTP    1-137
JCXPGS    1-127
JCXSMS    1-176
JOBCOM   45-6       45-13     45-23#
JS$OFF    1-65      42-157
JSTKND    1-87
JSWLOC    1-55
K52       1-68
KBMSG     1-158
KBTX      1-162
KCSIBF    1-147      8-63      8-81      8-84#     8-87#     8-89      8-102     8-103     8-104
KCSIMS    1-149      8-67
KDDCIN    1-26
KED       1-68
KILEMT    1-170     42-181     45-5      46-49
KL3CLR    1-75
KL4CLR    1-112
KMNBAS    1-133
KMNCHN    1-81      27-58
KMNNAM    1-174
KMPRMT    1-123
L         5-6        5-6#      5-7       5-7#      5-8       5-8#      5-9       5-9#      5-10      5-10#     5-11      5-11#
          5-12       5-12#     5-13      5-13#     5-14      5-14#     5-15      5-15#     5-16      5-16#     5-17      5-17#
          5-18       5-18#     5-19      5-19#     5-20      5-20#     5-21      5-21#     5-22      5-22#     5-23      5-23#
          5-30       5-30#     5-31      5-31#     5-38      5-38#     5-39      5-39#     5-40      5-40#     5-41      5-41#
          5-48       5-48#     5-49      5-49#     5-56      5-56#     5-57      5-57#     5-64      5-64#     6-7       6-7#
          6-8        6-8#      6-9       6-9#      6-10      6-10#     6-11      6-11#     6-12      6-12#     6-13      6-13#
          6-14       6-14#     6-15      6-15#     6-16      6-16#     6-17      6-17#
LAFSIZ    1-71
LBSPRI    1-138
LCDTYP    1-54
LCLUNT    1-50
LCOL      1-118
LCONTM    1-79      42-136
LCPUHI    1-79      42-74
LCPULO    1-79      42-75
LD$RON    1-107     23-74
LDBASE    1-108     26-85
LDCLEN    1-62      23-116    25-81
LDDEVN   23-13      23-46     24-35#
LDDEVX    1-110     18-65     26-66     26-76
LDFLAG    1-108     23-75#
LDMNT     1-59      23-80
LDNAM     1-147
LDNAME    1-108     23-62     23-89     25-75#
LDOPHD    1-150
LDPDEV    1-108     25-72#    26-84
LDSIZE    1-108     26-87
LF        2-4#       7-29     33-48     33-85     42-91     46-39
LF$OPN    1-140
LF$WRT    1-140     42-70
LFWLIM    1-71
LICTXT    1-129
LINBUF    1-69
```

Cross reference table (CREF V05.04)

```
LINCUR    1-71
LINFRE    1-169     37-41
LINIR     1-53
LINNXT    1-69
LINRTS    1-53
LITIME    1-90
LJSW      1-87
LMONHD    1-65
LMXLN     1-136
LMXPRM    1-137
LNBLKS    1-92
LNMAP     1-99
LNPRIM    1-99      33-37     33-51     33-121    42-80
LNSBLK    1-93
LNSPAC    1-103
LNUMER   33-31      33-34#    33-37     33-91
LOCKTX    1-158
LOFSPC    1-96      42-43     42-47     42-51*    42-59*
LOGASN    1-150     18-44     23-41
LOGBAS    1-117      1-119
LOGBLK    1-139
LOGBUF    1-139
LOGCHK    1-118     23-21     25-43     26-123
LOGCHN    1-139
LOGCLS    1-154     42-152
LOGDVU    1-117      1-119
LOGFLG    1-139     42-70*
LOGPTR    1-139
LOMAP     1-127
LOUTIR    1-53
LPARNT    1-57      34-41     34-43
LPRI      1-138
LPROG     1-78      42-119    42-201*
LPROJ     1-78      42-105    42-200*
LRBFIL    1-100
LSCCA     1-96
LSECPT    1-82      43-25
LSTACT    1-69
LSTATE    1-125
LSTDL     1-77      37-26     42-82     42-171
LSTHL     1-50
LSTIOL    1-50
LSTPL     1-120     33-119    42-15     42-169
LSTPRM    1-116
LSTSL     1-125     33-30     42-178    45-27     46-46
LSTSPL    1-52      32-43
LSUCF     1-74
LSW       1-49      33-36     33-97     33-122    33-124    42-58*    42-78
LSW11     1-38
LSW2      1-89
LSW2S     1-95
LSW3      1-95      42-21     42-31*    42-39*    42-81*
LSW4      1-112      7-21     33-103    33-132    42-56*
LSW5      1-75       7-15     42-19
LSW6      1-134      7-17*     7-34*
LSW7      1-137     33-101    33-128
```

Cross reference table (CREF V05.04)

```
LSW8      1-109
LSW9      1-73    35-5    42-23   42-57*  42-168*  42-196*
LTSCMD    1-103
LUNAME    1-78    33-57
LWINDO    1-38    42-84
MAXALC    1-51
MAXASN    1-91
MAXAVL    1-152
MAXMEM    1-55
MAXMTX    1-164
MAXPRI    1-55     1-138
MAXSEC    1-82    43-76
MHNSIZ    1-83
MHNSMS    1-84
MINTIM    1-82    42-135
MISSEQ    1-153
MNBASE    1-163   28-12*
MNBPC     1-162   28-8*   28-22*
MNFLGS    1-162   28-7*   28-31*
MNTARG    1-173   23-110  26-61
MNTCSH   22-58    23-99   23-110#
MNTDEV    1-148   22-34*  22-41   23-20   23-105   25-38*  25-42   25-49   25-65   25-76   26-36*
MNTFUL    1-150   24-15
MNTLD    22-57    23-68
MNTOP     1-163   28-16*
MNTTXT    1-179
MONAR1    1-163   28-37
MONAR2    1-163   28-49
MONHD     1-163    5-55#  28-27
MONVEC    1-138
MSGBUF    1-168   33-46
MSGEND    1-168   33-81
MTOPHD    1-149    5-79#  22-18
MUL32     1-178   40-78   40-43   42-99
MXJADR    1-58
MXJMEM    1-57
MXJPRI    1-138
MXPRMT    1-123
MXVEC     1-136
NAMTOP    1-129
NARGS     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    5-29#   5-30    5-30    5-31    5-31    5-37#   5-38    5-38    5-39    5-39    5-40
          5-40    5-41    5-41    5-47#   5-48    5-48    5-49    5-49    5-55#   5-56    5-56    5-57
          5-57    5-63#   5-64    5-64    6-6#    6-7     6-7     6-8     6-8     6-9     6-9     6-10
          6-10    6-11    6-11    6-12    6-12    6-13    6-13    6-14    6-14    6-15    6-15    6-16
          6-16    6-17    6-17
NATXT     1-156
NEDCHR    1-53
NEWJSW    1-87
NFRESB    1-114   32-10
NINTXT    1-156
NLINES    1-101
NMSGBF    1-184   33-114  33-140
NOCIN     1-147
```

Cross reference table (CREF V05.04)

```
NODAT     1-161
NOFIL     1-159     1-167     32-39
NOFLAG    1-149     22-12*    22-22*    23-72
NOFRDL    1-178     36-33
NOIND     1-153
NOLDMT    1-161
NONEMS    1-161
NOOPTT    1-167     33-16     34-37
NOPMGN    1-163     28-42
NOPRG     1-143     1-147
NOSTRT    1-146
NOTAVL    1-162
NOTIM    42-106    42-116    42-152#
NOTON     1-168     33-99
NOTXT     1-156
NOUDC     1-160
NSWPMS    1-164
NUCHN     1-100
NUMDEV    1-130     36-45
NUMON     1-71      46-21
NUMTB1    1-168     33-53
OCTFIX    1-156
OCTPRT    1-176
ODTBAS    1-133
OF$$SZ    1-132     35-20     35-96
OF$DEV    1-131     35-30*    35-54*    35-105*
OF$FIL    1-131     35-18     35-27*    35-28*    35-29*    35-70*    35-71*    35-77*    35-104*
OF$FLG    1-131     35-88*
OF$UNT    1-131     35-31*    35-55*    35-106*
OFFEMT    1-170     42-202
OFFHD     5-63#     42-10
OFFNWF    4-18#     42-9*     42-17     43-5*
OFFNWN    5-64      43-5#
OKFEND    1-83      35-15
OKFILE    1-83      35-11
OPRCMD    1-31      33-14#
OPTLST    1-46      21-15     25-17     42-11
OT$RON    1-132     35-88
OTHRON    1-172     46-23
OTRMNT    1-174     26-56
OVRCOR    1-146
PO$ALC    1-44      11-20     12-6
PO$BYP    1-44      26-14
PO$DBG    1-45
PO$DET    1-56      37-6
PO$LOK    1-54
PO$MEM    1-44      1-66
PO$NFW    1-44      26-14
PO$SND    1-56      33-43
PO$SYS    1-39      35-7
P2$WRL    1-56
PASLIN    1-119
PAUMSG    1-142     7-27
PBFEND    1-104
PF$IOW    1-134     5-56
PF$SYS    1-134     5-57
```

Cross reference table (CREF V05.04)

```
PFCO      1-45      16-38
PFSO      1-45      16-34
PHYMEM    1-86
PLOAD     1-27
PMBUSY    1-163     28-44
PNAME     1-130      1-152    36-46
POPCF     1-153     42-50
PRGALL    1-27      42-36
PRGSIZ    1-69
PRGTOP    1-69
PRIVCO    1-45      11-20     12-6      26-14     33-43     35-7      37-6      42-64
PRIVSO    1-44      42-65
PRMBUF    1-116
PRMEND    1-116
PRMPNT    1-115
PRTBUF    1-161
PRTDAT    1-177
PRTDC2    1-158
PRTDC3    1-158
PRTDEC    1-152     11-70     12-52     22-49     28-46     38-27     43-43
PRTFIX    1-155
PRTFNM    1-161
PRTLN     1-157
PRTPCT    1-171
PRTR50    1-177
PRTSPC    1-155
PRTTIM    1-177     42-90
PRTTMD    1-160
PRTTMV    1-159
PRTTOD    1-177
PRTTTP    1-156
PRTUNM    1-157
PRTWRN    1-191     24-15     33-34     33-114    33-140    35-102
PRVOPT    1-47       5-18
PUSHCF    1-144     42-45
PVNPW     1-45      16-40     42-66
PVON      1-75      42-173
QHDMS1    1-64      32-41
QHDMS2    1-64      32-42
R.GSIZ    1-60      14-12*
R.GSTS    1-61      14-13*    14-17     14-23*    14-25     14-36*
R.NAME    1-60      14-14*    14-15*
R50ADD    4-10#     15-13
R50BUF    1-148     10-21     11-38     11-85     12-42     13-23     13-29     14-14     14-15     15-12     22-30     23-104
          25-29     26-35     29-20     29-23*    29-28     29-36     35-42     35-47*    35-51     35-66     35-71     35-77
          39-25     46-15
R50COM    1-96      38-7
R50DEL    4-11#     15-15
R50DIR    1-144
R50DK     1-177     36-35
R50DSK    1-150     23-32
R50DUP    1-145
R50IND    1-144
R50K52    1-145
R50KED    1-145
R50KEX    1-145
```

Cross reference table (CREF V05.04)

```
R50LD     1-150     22-31     24-35     25-35
R50LDO    1-148     22-33     24-37     24-38     24-42     25-37     25-66     25-70
R50LD7    1-150     24-40     25-68
R50LOG    1-154
R50MON    1-178     39-27
R50NO     1-151
R50PIP    1-145
R50REM    4-12#     15-17
R50SAV    1-144     18-30
R50SET    4-7#
R50SY     1-144     36-33
R50TSX    1-145
R50TT     1-173
R50UCL    1-145
R50UP     4-8#
R50VIR    1-146
RDB       1-124
RDBEND    1-124
RDCMD     1-142      7-38      7-51      8-109     10-16     11-91     12-70     13-50     15-31     16-52     17-19     22-51
         23-120     25-82     28-48     28-51     30-25     30-32     31-22     31-74     31-86     31-97     32-44     32-50
         32-56      32-65     32-74     32-92     33-35     33-100    33-108    33-115    33-143    35-14     35-115    37-42
         39-47      40-47     41-19     42-27     42-60     45-45     46-41     47-9
RDERM     1-146
REMRDB    4-9#      14-11
REMRGN   13-35      14-7#
RESDEV    1-132     35-20*    35-103*   42-32*
RJEMT     1-176     45-19
RNMS      1-157
RONTXT    1-162
RS.CRR    1-60      14-17     14-25
RS.EGR    1-60      14-36
RS.GBL    1-60      14-13     14-23
RS.PVT    1-60      14-13
RSTPRV    1-61
RSTVEC   41-13      41-23#
RT$$SZ    1-124
RT$NAM    1-124
RUNMS     1-178     37-39
S$MSWT    1-81
SCHAIN    1-110
SD$BAK    1-112     31-96
SD$DEL    1-105     30-31
SD$FLK    1-106     31-46     31-51     32-31
SD$HLD    1-113
SD$SNG    1-114     32-14     32-49     32-55
SD$WFM    1-106     31-71     32-8
SDBLK     1-107
SDBU      1-112     31-94
SDBUF1    1-107
SDCB      1-120     29-31
SDCBND    1-120     29-32
SDCBSZ    1-125     29-38
SDDVU     1-124
SDFHD     1-42      32-37
SDFLAG    1-106     30-31*    31-45*    31-51*    31-71*    31-96*    32-8      32-14     32-31     32-49*    32-55*
SDFORM    1-106     31-53     32-24
```

Cross reference table (CREF V05.04)

```
SDNAME    1-48      29-34     29-36     31-17
SDSFCB    1-105     30-29     31-79     31-91     32-4
SDSKIP    1-112     31-85*
SEARCH    1-142     22-19     28-28     29-53     37-17
SEGCHN    1-38      27-56
SERFLG    1-49      23-25*    23-28*    23-90*    23-92*    25-58*    25-61*    42-46*    42-48*
SETHD     1-151
SFFORM    1-114
SFNMBL    1-114
SHMTH1    1-160
SHMTH2    1-160
SHOHD     1-175
SHTMSG    1-170     42-89
SIZEMT    1-173
SIZVAL    1-147
SJEMT     1-176     45-12
SKPD1    36-70#     36-72     36-74
SKPD2    35-97      36-71#
SKPDLM    1-37      11-35     12-37
SKPSPC    1-37      11-13     11-26     11-75     12-14     12-23     12-32     13-12     18-19     22-55     25-10     25-21
         26-30      34-14     34-24
SLMXLN    1-181
SMONHD    1-65
SMRSIZ    1-128
SNDALL   33-92      33-117#
SNDCOM   33-8       33-22#
SNDGN    33-15      33-30#
SNDLIN   33-27      34-10#
SNDONE   33-97#
SOPALC    1-52      11-16
SOPDAT    1-54      39-8
SOPTIM    1-54      40-8
SPACTV    1-165      1-166    32-6
SPCF      1-167     32-22
SPDLCR   30-10      30-29#
SPFLK     1-167     32-33
SPFUL     1-167     32-20
SPGEMT    1-167     32-101*   32-102
SPHLEM    4-30#     32-62*    32-63
SPLACT    1-170     42-189    46-26
SPLALN    6-7       31-5#
SPLBAK    6-8       31-90#
SPLBK1   31-92      31-94#
SPLCHN    1-80      27-57
SPLDEL    6-9       30-9#
SPLFRM    6-10      31-50#
SPLHD     1-164      6-6#     29-52
SPLHLA    1-155
SPLHLD    6-11      32-61#
SPLLK     6-13      31-44#
SPLMT1   31-46      31-52#
SPLMUL    6-14      32-54#
SPLND     1-107
SPLNHL    6-12      32-70#
SPLPND    1-172     46-28
SPLSKP    6-15      31-70#
```

Cross reference table (CREF V05.04)

```
SPLSNG    6-16      32-48#
SPLSTA    6-17      32-4#
SPNBB    31-80      31-93#
SPNBSY   30-30      30-33#    31-93
SPNHEM    4-36#     32-71*    32-72
SPOLGO   31-73      32-100#
SPSNG     1-165      1-167    32-16
SPUBUF    1-58
SPWFM     1-165      1-166    32-10
SQZDEV    4-5#      26-37*    26-74     26-122
SRTSIZ    1-128
SRTSMS    1-172
SRTTXT    1-175
SSRMAP    1-175
STLGCN    1-27
STLGHD    1-154
STPASK    1-172     46-33
STPFLG    1-80      42-87     42-162    46-45*    47-7*
SUBARG    1-162
SUBTXT    1-175
SUPCOD    1-129
SWPCHN    1-38      27-55
SWPTX     1-158
SXBPNT    1-58
SYFLVC   27-24      27-55#
SYHD1     1-157
SYHD2     1-157
SYINDX    1-130     36-37
SYNAME    1-131
SYSAV     1-142
SYSDAT    1-126     39-46*
SYTIMH    1-126     40-45*
SYTIML    1-126     40-46*
SYUNIT    1-130     36-40
TAB       2-8#
TALEMT    1-76      22-42
TBLOVF    1-151     35-17
TECO      1-68
TIMSPL    4-42#     42-191
TIMUP    42-92      42-105#
TK1SEC    1-128     42-96
TK1VAL    1-126
TM$SA1    1-37      43-40
TM$SA2    1-37      43-48
TM$SA3    1-37      43-42
TM$SA4    1-37      43-44
TMIDLH    1-62      41-29
TMIOH     1-62      41-26
TMIOWH    1-59      41-28
TMSWPH    1-62      41-27
TMSWTH    1-62      41-25
TMTOTH    1-59       1-171    41-23
TMTOTL    1-59       1-171
TMUSRH    1-59      41-24
TUOLNG    1-42
TOTMMS    1-175
```

Cross reference table (CREF V05.04)

```
TOTON     1-80      42-197
TOTXT     1-154
TRGRET    1-129
TRMSTR    1-144
TSKM2A    1-5#       1-26
TSXLN     1-129
TSXSIT    1-129
TSXSMS    1-176
UC$MDC    1-141
UC$NDC    1-141
UCHAN     1-98
UCISPC    1-83
UCLBLK    1-140
UCLDAT    1-140
UCLNAM    1-103
UERSEV    1-119
UFORM     1-89      32-81      32-85      32-88
UFPTRP    1-105
UHIMEM    1-92
UKMNAM    1-73
UMSSMS    1-175
UMSYTP    1-77
UPTMMS    1-170
USPLCH    1-80
USRMMS    1-176
USRSTK    1-57
USTART    1-88
UTRPAD    1-55
VCORTM    1-123
VCSHNB    1-82
VHIPCT    1-121
VIMAGE    1-87
VINTIO    1-113
VLDSYS    1-181     23-6
VQUAN0    1-120      1-121
VQUAN1    1-121
VQUAN2    1-121
VQUAN3    1-120      1-121
VQUN1A    1-121
VQUN1B    1-113
VQUN1C    1-113
VSWPFL    1-103
WILDFL    1-67
WLDNAM    1-42       2-11#     18-24     18-69     18-71     35-26     35-43     35-68
XAREA     1-143     14-16      14-24     14-37     23-91     31-18     31-27     31-32     31-34     38-11     42-47     42-54
         42-113     42-120     42-147    42-191
YELEMT    1-169     33-7*      33-21*    33-105    33-109    33-130    33-134
YELFLG    4-19#
YESTXT    1-156
```

Cross reference table (CREF V05.04)

```
...CM0    8-63    8-63    8-63   31-11   31-11   31-11   43-48   43-48   43-48   43-48
...CM1   23-91   31-18   31-27   31-32   31-34   38-11   42-47   42-54   42-113  42-120  42-147
...CM2   14-16   14-24   14-37   23-91   23-91   31-18   31-18   31-18   31-27   31-27   31-27   31-27
         31-32   31-32   31-32   31-32   31-34   31-34   31-34   31-34   38-11   38-11   42-47   42-47
         42-54   42-54   42-54   42-54   42-113  42-113  42-120  42-120  42-120  42-120  42-147  42-147
         42-147  42-147  42-191
...CM3   31-21   31-38   31-39   38-13   42-148
...CM5    7-23    7-27    7-50   11-71   12-53   14-16   14-24   14-37   22-50   23-91   25-54   28-47
         30-33   31-18   31-20   31-27   31-32   31-34   32-6    32-10   32-12   32-16   32-20   32-22
         32-27   32-33   32-35   32-39   32-41   32-42   33-99   33-107  37-39   37-41   37-49   38-11
         38-26   38-28   42-47   42-54   42-86   42-89   42-91   42-113  42-115  42-120  42-147  42-191
         43-40   43-42   43-44   46-23   46-28   46-33
...CM6   14-16   14-24   14-37   42-191
...CM7   31-27   31-32   31-34   42-54   42-120  42-147
.CLOSE    1-20#  31-21   31-38   31-39   38-13   42-148
.CRRG     1-16#  14-16   14-24
.CSIGE    1-18#  31-11
.CSISP    1-16#   8-63
.DATE     1-19#
.ELRG     1-16#  14-37
.ENTER    1-21#  31-18
.EXIT     1-21#
.FPROT    1-22#  23-91
.GTIM     1-19#
.GTLIN    1-19#  43-48
.GVAL     1-22#
.HERR     1-22#
.LOOKU    1-20#  38-11   42-47   42-113
.PRINT    1-20#   7-23    7-27    7-50   11-71   12-53   22-50   25-54   28-47   30-33   31-20   32-6
         32-10   32-12   32-16   32-20   32-22   32-33   32-35   32-39   32-41   32-42   33-99   33-107
         37-39   37-41   37-49   38-26   38-28   42-86   42-89   42-115  43-40   43-42   43-44   46-23
         46-28   46-33
.PURGE    1-17#
.PVAL     1-22#
.READW    1-17#  31-32   42-54   42-120
.REOPE    1-18#
.SAVES    1-18#
.SERR     1-22#
.SPFUN    1-19#
.SRESE    1-16#
.TTOUT    1-16#
.TTYIN    1-17#   7-28   46-34
.TTYOU    1-17#  32-27   42-91
.TWAIT    1-17#  42-191
.WRITW    1-21#  31-27   31-34   42-147
CMDDEF    3-47#   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    5-30    5-31    5-38    5-39    5-40
          5-41    5-48    5-49    5-56    5-57    5-64    6-7     6-8     6-9     6-10    6-11    6-12
          6-13    6-14    6-15    6-16    6-17
FABORT    3-15#   8-70    8-97   11-22   11-63   11-66   12-8    12-26   12-57   13-46   15-27   17-10
         18-57   18-58   18-99   19-19   22-21   22-28   23-8    23-15   23-36   23-50   23-82   25-24
         26-16   26-33   26-44   26-56   28-30   28-42   28-55   29-43   29-65   29-66   30-24   31-8
         33-16   33-45   34-19   34-37   35-9    35-17   35-119  37-8    37-19   37-25   38-33   38-37
         38-41   39-50   40-50   45-29
FERR      3-4#   11-59   11-69   12-51   14-30   14-42   22-48   24-17   28-44   42-52
FWARN     3-23#  24-15   33-34   33-114  33-140  35-102
```

Cross reference table (CREF V05.04)

TBLDEF	3-35#	5-5	5-29	5-37	5-47	5-55	5-63	6-6
TBLEND	3-61#	5-24	5-32	5-42	5-50	5-58	5-65	6-18