TSKM2A -- Keyboard command rout MACRO V05.05 Thursday 19-Jan-89 14:53
Table of contents


```
    4-    1       Data areas
    5-    1       Tables of commands and options
    7-    1       PAUSE command
    7-   40       DISPLAY command
    8-    1       ASSIGN command
   10-    1       DEASSIGN command
   11-    1       ALLOCATE command
   12-    1       DEALLOCATE command
   13-    1       REMOVE command
   15-    1       INSTALL Command
   22-    1       MOUNT command
   25-    1       DISMOUNT command
   26-    1       INITIALIZE command
   26-    2       SQUEEZE command
   26-    3       FORMAT command
   28-    1       MONITOR command
   29-    1       SPOOL command
   33-    1       SEND command
   35-    1       DETACH command
   37-    1       DATE command
   38-    1       TIME command
   39-    1       RESET command
   40-    1       OFF command
   43-    1       KILL command
   43-    8       SUSPEND command
   43-   15       RESUME command
   44-    1       BOOT and $STOP commands
   45-    1       $SHUTDOWN command
```

```
                                        .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, 1985, 1986, 1987, 1988.
                                ;  S&H Computer Systems, Inc.
                                ;  Nashville, Tennessee
                                ;
                                ;  Macro calls
                                ;
                                        .MCALL  .CSISPC,.CRRG,.ELRG
                                        .MCALL  .READW,.TTYIN,.TTYOUT,.TWAIT
                                        .MCALL  .CSIGEN
                                        .MCALL  .GTLIN
                                        .MCALL  .PRINT,.CLOSE,.LOOKUP,.CSTAT
                                        .MCALL  .WRITW,.ENTER
                                        .MCALL  .FPROT
                                ;
                                ;  Global definitions
                                ;
                                        .GLOBL  TSKM2A,CMDOFF
                                        .GLOBL  CMDFRM,CMDDSN,PRGALL
                                        .GLOBL  DLCEMT,ALCDEV,CMDINS,CMDSPN,CMDRSM,CMDYEL
                                        .GLOBL  CMDRST,CMDSND,CMDFMT,CMDDAT
                                        .GLOBL  CMDTIM,CMDMNT,CMDMON,CMDDMT,CMDDSP,CMDASN
                                        .GLOBL  CMDALC,CMDDLC,OPRCMD,CMDSPO,CMDKIL,CMDREM
                                        .GLOBL  CMDPAU,CMDDET,CMDINI,CMDSQZ
                                        .GLOBL  CMDBOT,CMDSHT
                                ;
                                ;  Global references
                                ;
                                        .GLOBL  TM$SA1,TM$SA2,TM$SA3,TM$SA4,SKPSPC,SKPDLM
                                        .GLOBL  SWPCHN,SEGCHN,INDTSV,LWINDO
                                        .GLOBL  $SCCA,AF$CCA,AF$NPW
                                        .GLOBL  AF$DUP,AF$IND,AF$UCL,AF$SET,FORCEO
                                        .GLOBL  CFSTOP,CFSTRT,EM$ALC,EM$NFW,EM$SND,EM$DNR
                                        .GLOBL  SDFHD,AF$TPO,EM$CSE,WLDNAM
                                        .GLOBL  AF$PLK,AF$DBG,EM$LDI,EM$NLN
                                        .GLOBL  PO$NFW,PO$BYP,PO$ALC,CKSYPV,CKACOJ,PRIVSO
                                        .GLOBL  PFSO,PFCO,PVNPW,PRIVCO
                                        .GLOBL  CLRPRV,EM$FNI,EM$ITF,INPADR,INPEMT,OPTLST
                                        .GLOBL  CDBUF,CDGET,INGADR,INGEMT,IIBUF,PRVOPT
                                        .GLOBL  $VNOTT,SDNAME,DMYDEV
                                        .GLOBL  CORUSR,LSW,SERFLG
                                        .GLOBL  LSTSPL,SOPALC
                                        .GLOBL  SOPDAT,SOPTIM
                                        .GLOBL  ERRLOC
                                        .GLOBL  PO$DET,EM$DET,PO$SND
                                        .GLOBL  LPARNT
                                        .GLOBL  TMTOTH,TMUSRH,TMIOWH,LDMNT
                                        .GLOBL  R.GSIZ,RS.GBL,RS.PVT,R.NAME,RS.CRR,RS.EGR
                                        .GLOBL  EM$UAR,EM$UER,R.GSTS
```

```
                        .GLOBL  TMSWTH, TMIDLH, TMIOH, TMSWPH, LDCLEN
                        .GLOBL  EM$IAD, EM$ATF, EM$NLD
                        .GLOBL  EM$SSY, EM$NID, QHDMS1, QHDMS2, EM$OLO
                        .GLOBL  GENMON, JS$OFF, AF$MEM, AF$BYA
                        .GLOBL  EM$DAA, EM$DIU, CHKALC
                        .GLOBL  $NOIN
                        .GLOBL  AF$SCA
                        .GLOBL  NUMON
                        .GLOBL  DOASGN, LSW9
                        .GLOBL  $PRGLK, LSW5, PVON
                        .GLOBL  ALDEMT, TALEMT
                        .GLOBL  LSTDL, FSTDL, $DETCH
                        .GLOBL  LPROJ, LPROG, LUNAME
                        .GLOBL  LCPUHI, LCPULO, LCONTM, $CTRLS
                        .GLOBL  STPFLG, TOTON, SPLCHN
                        .GLOBL  CFBUF, CCLSAV, KMNCHN
                        .GLOBL  MINTIM, LSECPT, MAXSEC
                        .GLOBL  CASTBR, CASCBR, CASTBW, CASCUP
                        .GLOBL  CASTRO, CASTWO
                        .GLOBL  CTRLTT
                        .GLOBL  BOTDEV
                        .GLOBL  $INKMN, UFORM
                        .GLOBL  $CFABT, INDERR
                        .GLOBL  ASNTBL, $DILUP, CSHDEV, CSHDVN
                        .GLOBL  AT$LOG, AT$SIZ, AT$DEV, AT$FIL, AT$$SZ, AT$EXT
                        .GLOBL  ASNEND, LSW3
                        .GLOBL  LOFSPC, R50COM
                        .GLOBL  EM$FOE
                        .GLOBL  LNPRIM, CW$50H, CONFIG
                        .GLOBL  $DOOFF
                        .GLOBL  C.CSW, C.DEVQ
                        .GLOBL  CD$DVU, CD$BAS, CD$$SZ
                        .GLOBL  $CFOPN, $TTGAG
                        .GLOBL  SDSFCB, SD$DEL
                        .GLOBL  SDFLAG, SD$FLK, SD$WFM, SDFORM, SD$FLG
                        .GLOBL  LD$RON
                        .GLOBL  LDNAME, LDSIZE, LDFLAG, LDBASE, LDPDEV
                        .GLOBL  CFCHAN, LDDEVX
                        .GLOBL  CFPNT, $QUIET
                        .GLOBL  LSW4, SDSKIP, SDBU, SD$BAK
                        .GLOBL  SD$SNG, NFRESB
                        .GLOBL  SD$WID
                        .GLOBL  CFSPND
                        .GLOBL  LOGCHK
                        .GLOBL  LSTPL, SDCB, SDCBND
                        .GLOBL  DCTRD, DCCRD, DCTWR, DCCWR, ASNSRC
                        .GLOBL  SDCBSZ, LSTSL
                        .GLOBL  SYSDAT, SYTIMH, SYTIML
                        .GLOBL  TK1SEC
                        .GLOBL  RESDEV
                        .GLOBL  LSW6, PF$SYS, PF$IOW
                        .GLOBL  INDSAV
                        .GLOBL  LSW7
                        .GLOBL  LOGFLG
                        .GLOBL  LF$WRT
                        .GLOBL  CVTUC
                        .GLOBL  CMDBUF, PAUMSG, RDCMD, CVTTAB, SEARCH
```

```
                                .GLOBL  INVOPT,FKILL,ABRTCF,INDABT,XAREA,FPRINT
                                .GLOBL  PUSHCF,FILNAM,R50SAV
                                .GLOBL  BLKO
                                .GLOBL  ASKLNM,BADCMD,KCSIBF
                                .GLOBL  ASDEX,KCSIMS,ASNOVF,GTRD50,R50BUF,R50LDO,MNTDEV,DMTARG
                                .GLOBL  DEADEV,CHKMNT,CHKMTX,INFOMT,NOFLAG,MTOPHD,ILLCMD
                                .GLOBL  R50LD,INVLDN,R50DSK,ACRFIL,BDFNAM,LOGASN,MNTFUL,R50LD7
                                .GLOBL  CKPRIV,AMBOPT,ACRDEC
                                .GLOBL  PRTDEC
                                .GLOBL  ACROCT,CSIMS1,POPCF
                                .GLOBL  CRLF,LOGCLS
                                .GLOBL  DIVIDE
                                .GLOBL  CPUAH,CPUAL
                                .GLOBL  MNFLGS,MNBPC
                                .GLOBL  DELSPC,MNBASE,MNTOP,MONHD,MONAR1,NOPMGN,PMBUSY,MONAR2
                                .GLOBL  CHKDLM,SPLHD
                                .GLOBL  DEVIDL,ALDEX,SPACTV,SPWFM,SPSNG
                                .GLOBL  COAL,ALDBLK,COAD,SPFLG
                                .GLOBL  SPFUL,SPCF,SPFLK,NOFIL,SPGEMT,NOOPTT,NSPFLG
                                .GLOBL  BDLIN,MSGBUF,NUMTB1,MSGEND,NOTON,GAGMSG,SPWIDE,SPNARG
                                .GLOBL  YELEMT,LINFRE,DJABMS,DLMSG,INVTIM,DMTALL
                                .GLOBL  SHTMSG,AUTHFN,SPLACT,DOSTOP,OFFEMT,KILEMT
                                .GLOBL  OTHRON,SPLPND,STPASK
                                .GLOBL  INVLDM,CSIMS4,MNTARG
                                .GLOBL  OTRMNT,CHKDEV,DMTSUB,CMDCCL
                                .GLOBL  SJEMT,RJEMT
                                .GLOBL  PRTTIM,INVDEV,ALFN
                                .GLOBL  DETHD,DETARG,RUNMS,NOFRDL,R50MON,INVDAT,MUL32,COAF
                                .GLOBL  AR$PRJ,AR$PRG,AR$CON,AR$CNT,AR$CPH,AR$CPL
                                .GLOBL  AR$$SZ,ARNRPB
                                .GLOBL  PRTWRN,VLDSYS,$LOFCF
                                .GLOBL  INSTBL,INSTBN,AF$NOW,AF$HIE,AF$NOI,II$$SZ
                                .GLOBL  AF$IOP,II$NAM,II$FLG,II$PRV,II$NPV
                                .GLOBL  NMSGBF,$NOABT,BADBOT
```

```
                        ;
                        ;   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'HD: .WORD  2*NA
                              .ENDM   TBLDEF
                        ;
                        ;-------------------------------------------------------------
                        ;  Macro to enter an option text name and a set of parameters
                        ;  into the currently open table.
                        ;  STRNG = Ascii name
                        ;  A,B,C = Set of option parameters to store in table with name.
                        ;
                              .MACRO  CMDDEF  STRNG,A,B,C
                              .CSECT  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                          CINFO:  .BLKW   6.              ;Cstat information necessary for boot
000020                          REMRDB: .BLKW   5
000032  003344                  R50ADD: .RAD50  /ADD/
000034  014724  021145          R50DEL: .RAD50  /DELETE/
000040  070525                  R50REM: .RAD50  /REM/
000042  000000                  IATTRB: .WORD   0
                                ;
                                ;  Byte data
                                ;
000044      000                 DMTNLG: .BYTE   0
000045      000                 OFFNWF: .BYTE   0
                                        .EVEN
                                ;
                                ;  Emt to delete a spool file
                                ;
000046      000     126         DELEMT: .BYTE   0,126
000050  000007                          .WORD   7
000052  000000                          .WORD   0               ;File ID goes here
                                ;
                                ;  Emt to set spool HOLD mode
                                ;
000054      000     126         SPHLEM: .BYTE   0,126
000056  000015                          .WORD   15
000060  000000                          .WORD   0               ;Address of SDCB
                                ;
                                ;  Emt to set spool NOHOLD mode
                                ;
000062      000     126         SPNHEM: .BYTE   0,126
000064  000016                          .WORD   16
000066  000000                          .WORD   0               ;Address of SDCB
                                ;
                                ;  Time to wait for check spooler activity.
                                ;
000070  000000  000170          TIMSPL: .WORD   0,60.*2.        ;2 second timer
```

```
                        .SBTTL  Tables of commands and options
                ;-----------------------------------------------------------
                ;  Define option switches for the INSTALL ADD command
                ;
000074                  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
                ;
000074                  TBLDEF  MTOP,1
000162                  CMDDEF  WR*ITE,0
000166                  CMDDEF  NOW*RITE,1
000172                  TBLEND

                ;-----------------------------------------------------------
                ;  Define option switches for the DISMOUNT command
                ;
000074                  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
                ;
000074                  TBLDEF  DET,1
000222                  CMDDEF  CH*ECK,DETCHK
000226                  CMDDEF  KI*LL,DETKIL
000232                  TBLEND

                ;-----------------------------------------------------------
                ;  Options for the MONITOR command.
                ;
000074                  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 000074                                         TBLDEF  OFF,1
      64 000252                                         CMDDEF  N*OWARN,OFFNWN
      65 000256                                         TBLEND
```

```
                                ;-----------------------------------------------------------------
                                ;  Define options for the SPOOL command
                                ;  Entry 1 = Option name
                                ;  Entry 2 = Processing routine
                                ;
000074                                TBLDEF  SPL,1
000262                                CMDDEF  A*LIGN,SPLALN
000266                                CMDDEF  B*ACKUP,SPLBAK
000272                                CMDDEF  D*ELETE,SPLDEL
000276                                CMDDEF  FL*AGPAGE,SPLFLG
000302                                CMDDEF  NOF*LAGPAGE,SPLNFL
000306                                CMDDEF  F*ORM,SPLFRM
000312                                CMDDEF  H*OLD,SPLHLD
000316                                CMDDEF  NOH*OLD,SPLNHL
000322                                CMDDEF  L*OCK,SPLLK
000326                                CMDDEF  M*ULTIPLE,SPLMUL
000332                                CMDDEF  NA*RROW,SPLNRW
000336                                CMDDEF  SK*IP,SPLSKP
000342                                CMDDEF  SI*NGLE,SPLSNG
000346                                CMDDEF  ST*ATUS,SPLSTA
000352                                CMDDEF  W*IDE,SPLWID
000356                                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.
                                ;
000074  016767  000000G 000000G CMDPAU: MOV     CFPNT,CFSPND    ;SUSPEND COMMAND FILE
000102  001441                        BEQ     9$              ;BR IF COMMAND FILE NOT RUNNING
000104  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.
                                ;
000110  032761  000000G 000000G         BIT     #$SCCA,LSW5(R1) ;Is control-C abort override in effect?
000116  001003                        BNE     3$              ;Br if yes -- Don't abort on control-C
000120  052761  000000G 000000G         BIS     #$CFABT,LSW6(R1);Abort command files if ctrl-C typed
                                ;
                                ;  If TTY QUIET is set, print pause message for operator
                                ;
000126  032761  000000G 000000G 3$:     BIT     #$QUIET,LSW4(R1);IS QUIET SET ON?
000134  001403                        BEQ     2$              ;BR IF NOT
000136                                  .PRINT  #CMDBUF         ;DISPLAY THE PAUSE COMMAND
                                ;
                                ;  Print pause message and wait for response
                                ;
000144                          2$:     .PRINT  #PAUMSG         ;PRINT PAUSE MESSAGE
000152                          1$:     .TTYIN                  ;ACCEPT A LINE OF INPUT
000156  120027  000012                  CMPB    R0,#LF
000162  001373                        BNE     1$
                                ;
                                ;  Restart the command file
                                ;
000164  042761  000000G 000000G         BIC     #$CFABT,LSW6(R1);Clear command-file abort flag
000172  016700  000000G                 MOV     CFSPND,R0       ;RESTART COMMAND FILE
000176  004767  000000G                 CALL    CFSTRT
000202  005067  000000G                 CLR     CFSPND
000206  000167  000000G          9$:    JMP     RDCMD

                                        .SBTTL  DISPLAY command
                                ;-------------------------------------------------------------------------
                                ;  Display a line from within a command file.
                                ;
000212  124327  000040          CMDDSP: CMPB    -(R3),#'        ;BACKUP OVER LEADING SPACES
000216  001775                        BEQ     CMDDSP
000220  005203                        INC     R3              ;POINT TO FIRST SEPARATOR
000222  122327  000040                  CMPB    (R3)+,#'        ;IS IT A SPACE?
000226  001401                        BEQ     1$              ;BR IF YES (SKIP IT)
000230  005303                        DEC     R3              ;POINT TO 1ST SEPARATOR
000232                          1$:     .PRINT  R3              ;PRINT THE ARGUMENT OF THE DISPLAY COMMAND
000236  000167  000000G                 JMP     RDCMD
```

```
                                        .SBTTL  ASSIGN command
                                ;-------------------------------------------------
                                ;  PROCESS THE 'ASSIGN' COMMAND.
                                ;
                                ;  SEE IF FULL FILE ID IS SPECIFIED OR JUST DEVICE NAME.
                                ;
000242  004767  000000G       CMDASN: CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACES
000246  105067  000000G               CLRB    ASKLNM          ;SET FLAG FOR LONG ASSIGN
000252  126227  177777  000072       CMPB    -1(R2),#':      ;DOES LOGICAL NAME HAVE COLON?
000260  001001                       BNE     5$              ;BR IF NOT
000262  105042                       CLRB    -(R2)           ;CLEAR IT FOR NOW
000264  010201                5$:    MOV     R2,R1           ;POINT TO NULL AT END OF LINE
000266  020103                1$:    CMP     R1,R3           ;SCAN BACK TOWARDS FRONT
000270  101002                       BHI     2$              ;BRANCH IF MORE TO DO
000272  000167  000000G               JMP     BADCMD          ;MUST HIT SOME DELIM
000276  114100                2$:    MOVB    -(R1),R0        ;GET NEXT CHAR BACK
000300  004767  000320               CALL    ASDELM          ;IS IT A DELIMITER?
000304  001370                       BNE     1$              ;IF NOT KEEP SCANNING
                                ;  WE SCANNED BACK TO A DELIMITER -- SEE WHERE IT IS
                                ;  IN COMMAND.
000306  120027  000072               CMPB    R0,#':          ;REPLACE ':' AND ' ' WITH '='
000312  001414                       BEQ     3$
000314  120027  000040               CMPB    R0,#' 
000320  001013                       BNE     4$
000322  020103                6$:    CMP     R1,R3           ;HAVE WE REACHED FRONT OF COMMAND LINE?
000324  101407                       BLOS    3$              ;BR IF YES
000326  124127  000040               CMPB    -(R1),#' '      ;SKIP OVER ALL CONSECUTIVE BLANKS
000332  001773                       BEQ     6$
000334  121127  000072               CMPB    (R1),#':        ;ALLOW "ASSIGN DX1: DK" CONSTRUCT
000340  001401                       BEQ     3$
000342  005201                       INC     R1              ;POINT TO 1ST BLANK CHARACTER
000344  112711  000075        3$:    MOVB    #'=,(R1)
000350  010100                4$:    MOV     R1,R0           ;HOW MANY CHARS IN NAME TO
000352  160300                       SUB     R3,R0           ;LEFT OF DELIMITER?
000354  020027  000003               CMP     R0,#3           ;DEVICE NAME <= 3 CHARS
000360  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.
                                ;
000362  112722  000072       ASSMPL: MOVB    #':,(R2)+       ;PUT COLON AT END OF 2ND NAME
000366  105022                       CLRB    (R2)+           ;NULL TO TERMINATE STRING
000370  010204                       MOV     R2,R4           ;MAKE ROOM AT END OF 1ST NAME
000372  062704  000002               ADD     #2,R4           ;BY SLIDING CHARS OVER 2 PLACES
000376  020201                1$:    CMP     R2,R1           ;WORK BACK DOWN TO '='
000400  001402                       BEQ     2$              ;BRANCH AFTER '=' MOVED
000402  114244                       MOVB    -(R2),-(R4)     ;SLIDE OVER TO RIGHT
000404  000774                       BR      1$
000406  112722  000072        2$:    MOVB    #':,(R2)+       ;PUT COLON AFTER 1ST DEVICE NAME
000412  112712  000101               MOVB    #'A,(R2)        ;PUT IN DUMMY FILE NAME 'A'.
000416  105267  000000G              INCB    ASKLNM          ;REMEMBER TO KILL FILE NAME LATER
000422  000403                       BR      ASCIS           ;GO DO .CSISPC
                                ;
                                ;  REQUEST ASSIGNMENT OF LOGICAL DEVICE TO NAMED FILE.
                                ;
000424  112722  000072       ASFID:  MOVB    #':,(R2)+       ;PUT COLON AT END OF LOGICAL NAME
```

```
                        CLRB    (R2)            ;NULL TO END STRING
                ;
                ;  CALL CSI IN SPECIAL MODE TO PARSE ASSIGN COMMAND.
                ;
ASCIS:  MOV     SP,R5           ;SAVE STACK POINTER
        .CSISPC #KCSIBF,#ASDEX,R3
        MOV     R5,SP           ;RESET TOP OF STACK
        BCC     ASCSOK          ;BRANCH IF .CSISPC OK
                ;  ERROR OCCURED ON .CSISPC
CSIERR: MOVB    @#ERRLOC,R5     ;GET ERROR CODE
        ASL     R5              ;CONVERT TO WORD INDEX
        MOV     KCSIMS(R5),R4   ;GET ADDR OF ERROR MESSAGE
        FABORT  R4              ;PRINT ERROR MESSAGE
                ;
                ;  Move the assign information into the assign table.
                ;  The physical device (and possibly file name)
                ;  description is in KCSIBF in the area for chan #0.
                ;  The logical device block is in the space for chan #3.
                ;
                ;  If "physical" device name used in assign statement is actually
                ;  a logical name (previously assigned), convert it to the corresponding
                ;  physical device name.
                ;
ASCSOK: MOV     KCSIBF,R0       ;GET PHYSICAL DEVICE NAME FOR ASSIGN
        CALL    ASNSRC          ;SEE IF IT IS ACTUALLY A LOGICAL DEV NAME
        BCS     3$              ;BR IF NOT
        MOV     AT$DEV(R2),KCSIBF;REPLACE LOGICAL NAME WITH REAL PHYSICAL NAME
3$:     TSTB    ASKLNM          ;MUST WE KILL FILE NAME?
        BEQ     1$              ;BRANCH IF NOT
        CLR     <KCSIBF+2>      ;KILL THE NAME
                ;  Clear any previous assign with this logical name.
1$:     MOV     KCSIBF+30.,R0   ;GET LOGICAL DEVICE NAME
        CALL    ASNSRC          ;SEE IF NAME IS ALREADY ASSIGNED
        BCC     2$              ;BR IF ASSIGN ALREADY EXISTS (REUSE THIS ENTRY
                ;  Search for a free assign block.
        CLR     R0              ;SEARCH FOR FREE ASSIGN BLOCK
        CALL    ASNSRC
        BCC     2$              ;BR IF FOUND ONE
                ;  ASSIGN TABLE IS FULL.
        FABORT  #ASNOVF
                ;
                ;  FOUND A FREE SLOT (POINTED TO BY R2) MOVE ASSIGN
                ;  INFO INTO BLOCK.
                ;
2$:     MOV     KCSIBF+30.,AT$LOG(R2);MOVE IN LOGICAL DEVICE NAME
        MOV     KCSIBF+8.,AT$SIZ(R2);MOVE IN FILE LENGTH
        MOV     #KCSIBF,R3
        MOV     (R3)+,AT$DEV(R2);MOVE IN REAL DEVICE NAME
        MOV     (R3)+,AT$FIL(R2);MOVE IN FILE NAME
        MOV     (R3)+,AT$FIL+2(R2)
        MOV     (R3),AT$EXT(R2) ;MOVE IN FILE EXTENSION
        JMP     RDCMD
```

```
      1                                ;-------------------------------------------------------
      2                                ;  ASDELM IS CALLED TO CHECK IF THE CHARACTER IN R0
      3                                ;  IS ONE OF THE CHARACTERS ':', '=', OR ' '.
      4                                ;  ON RETURN THE CONDITION CODE IS SET FOR BEQ/BNE.
      5                                ;  ALL REGISTERS ARE PRESERVED.
      6                                ;
      7 000624  120027  000075         ASDELM: CMPB    R0,#'=          ;CHECK EQUAL SIGN
      8 000630  001405                         BEQ     1$
      9 000632  120027  000072                 CMPB    R0,#':          ;CHECK COLON
     10 000636  001402                         BEQ     1$
     11 000640  120027  000040                 CMPB    R0,#'           ;CHECK SPACE
     12 000644  000207                 1$:     RETURN          ;RETURN WITH CC SET.
```

```
                                        .SBTTL  DEASSIGN command
                                ;-------------------------------------------------------------
                                ;  PROCESS THE DEASSIGN COMMAND
                                ;
000646  004767  000000G         CMDDSN: CALL    CVTTAB          ;CONVERT TAB AND FF TO SPACES
000652  105713                          TSTB    (R3)            ;DEASSIGN ALL OR ONE?
000654  001010                          BNE     DEAS1           ;BR TO DEASSIGN ONE NAME
                                ;
                                ;  REQUEST TO DEASSIGN ALL LOGICAL NAMES
                                ;  (SIMPLY ZERO THE ASSIGN TABLE)
                                ;
000656  012702  000000G                 MOV     #ASNTBL,R2      ;POINT TO START OF TABLE
000662  005022                  1$:     CLR     (R2)+
000664  020227  000000G                 CMP     R2,#ASNEND      ;DONE ALL?
000670  103774                          BLO     1$              ;BR IF NOT
000672  000167  000000G         DAJMP:  JMP     RDCMD
                                ;
                                ;  REQUEST TO DEASSIGN A PARTICULAR LOGICAL DEVICE NAME
                                ;
000676  004767  000000G         DEAS1:  CALL    GTRD50          ;ACCRUE THE DEVICE NAME
000702  016700  000000G                 MOV     R50BUF,R0       ;PICK UP DEVICE NAME
000706  012702  000000G                 MOV     #ASNTBL,R2      ;POINT TO START OF ASSIGN TABLE
000712  020062  000000G         1$:     CMP     R0,AT$LOG(R2)   ;IS THIS ENTRY FOR THE DEVICE?
000716  001004                          BNE     2$              ;BR IF NOT
000720  005062  000000G                 CLR     AT$LOG(R2)      ;DEASSIGN THE LOGICAL NAME
000724  005062  000000G                 CLR     AT$DEV(R2)      ;CLEAR PHYSICAL DEVICE NAME ALSO
000730  062702  000000G         2$:     ADD     #AT$$SZ,R2      ;POINT TO NEXT ASSIGN TABLE ENTRY
000734  020227  000000G                 CMP     R2,#ASNEND      ;REACHED END OF TABLE?
000740  103764                          BLO     1$              ;BR IF MORE TO DO
000742  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
                ;
000744          CMDALC:
                ;
                ;  If no devices were specified, treat this like a SHOW ALLOCATE command
                ;
000744  005067  177032          CLR     ALC1DV          ;Say no allocat device seen yet
000750  004767  000000G         CALL    CVTTAB          ;Convert tab and FF chars to spaces
000754  004767  000000G         CALL    SKPSPC          ;Skip over spaces
000760  105713                  TSTB    (R3)            ;Any devices specified?
000762  001002                  BNE     1$              ;Br if yes
000764  000167  000000G         JMP     SOPALC          ;Do SHOW ALLOCATE command
                ;
                ;  See if we are authorized to allocate devices
                ;
000770  032767  000000G 000000G 1$:     BIT     #PO$ALC,PRIVCO  ;Are we authorized to allocate devices?
000776  001004                  BNE     10$             ;Br if yes
001000                          FABORT  #EM$ALC         ;Not authorized for ALLOCATE
                ;
                ;  Begin loop to get the name of each device to be allocated
                ;
001010  004767  000000G         10$:    CALL    SKPSPC          ;Skip over any spaces
                ;
                ;  Accrue the next device name
                ;
001014  004767  000000G                 CALL    GTRD50          ;Accrue the device name
001020  122327  000072                  CMPB    (R3)+,#':       ;Was a colon specified with the device name?
001024  001401                          BEQ     6$              ;Br if yes
001026  005303                          DEC     R3              ;Backup pointer
001030  010346                  6$:     MOV     R3,-(SP)        ;Save command pointer
001032  004767  000000G                 CALL    SKPDLM          ;Skip legal delimiters (blank(s), comma, EOL)
001036  012603                          MOV     (SP)+,R3        ;Restore command pointer
001040  103435                          BCS     41$             ;Invalid character following name
001042  016702  000000G                 MOV     R50BUF,R2       ;Get the name of the device
001046  010267  000000G                 MOV     R2,ALCDEV       ;Set name of device being allocated
                ;
                ;  Save the first device name
                ;
001052  005767  176724                  TST     ALC1DV          ;Have we seen the first device yet?
001056  001002                          BNE     2$              ;Br if yes
001060  010267  176716                  MOV     R2,ALC1DV       ;Save the name of the 1st device
                ;
                ;  Do the allocation for this device
                ;
001064  012700  000000G         2$:     MOV     #ALDEMT,R0      ;Point to EMT arg block to do the allocation
001070  104375                          EMT     375             ;Try to allocate the device
001072  103051                          BCC     3$              ;Br if allocation was successful
001074  010005                          MOV     R0,R5           ;Save returned error value
                ;
                ;  An error occurred on the allocation.  Print error message.
                ;
001076  113702  000000G                 MOVB    @#ERRLOC,R2     ;Get allocation error code
001102  120227  000001                  CMPB    R2,#1           ;Device already allocated by someone else?
```

```
     58 001106  001007                        BNE     4$              ;Br if not
     59 001110                                FERR    #EM$DAA         ;Device allocated by another job
     60 001124  000427                        BR      7$
     61 001126  120227  000002        4$:     CMPB    R2,#2           ;Invalid device being allocated?
     62 001132  001004                        BNE     5$              ;Br if not
     63 001134                        41$:    FABORT  #EM$IAD         ;Invalid device for allocation
     64 001144  120227  000003        5$:     CMPB    R2,#3           ;Allocation table full?
     65 001150  001004                        BNE     8$              ;Br if not
     66 001152                                FABORT  #EM$ATF         ;Allocation table full
     67 001162  120227  000004        8$:     CMPB    R2,#4           ;Device in use by another user?
     68 001166  001013                        BNE     3$              ;Br if not
     69 001170                                FERR    #EM$DIU         ;Device is in use by another user
     70 001204  004767  0000006       7$:     CALL    PRTDEC          ;Print the number of the job that has the dev
     71 001210                                .PRINT  #CRLF           ;End of line
     72                                ;
     73                                ;  See if there are more devices to be allocated
     74                                ;
     75 001216  004767  0000006       3$:     CALL    SKPSPC          ;Skip over any spaces
     76 001222  122327  000054                CMPB    (R3)+,#',       ;Is there another device to allocate?
     77 001226  001670                        BEQ     10$             ;Loop if yes
     78                                ;
     79                                ;  We have finished allocating all devices.
     80                                ;  See if we need to assign a logical name to the first device we allocated.
     81                                ;
     82 001230  105743                        TSTB    -(R3)           ;Was a logical device name specified?
     83 001232  001410                        BEQ     9$              ;Br if not
     84 001234  004767  0000006               CALL    GTRD50          ;Accrue the logical device name
     85 001240  016705  0000006               MOV     R50BUF,R5       ;Get the logical device name
     86 001244  016700  176532                MOV     ALC1DV,R0       ;Get the physical device name
     87 001250  004767  0000006               CALL    DOASGN          ;Do the assignment
     88                                ;
     89                                ;  Finished
     90                                ;
     91 001254  000167  0000006       9$:     JMP     RDCMD           ;Finished with command
```

```
                                        .SBTTL  DEALLOCATE command
                                ;-------------------------------------------------------------------------------
                                ;  Deallocate an allocated device.
                                ;
001260  005067  176516          CMDDLC: CLR     ALC1DV          ;Clear deallocated device counter
001264  032767  000000G 000000G        BIT     #PO$ALC,PRIVCO  ;Are we authorized to allocate devices?
001272  001004                        BNE     1$              ;Br if yes
001274                                FABORT  #EM$ALC         ;Not authorized to deallocate
                                ;
                                ;  See if /ALL was specified
                                ;
001304  004767  000000G         1$:    CALL    CVTTAB          ;Convert tab and FF chars to spaces
001310  004767  000000G                CALL    CVTUC           ;Convert lower case chars to upper case
001314  004767  000000G                CALL    SKPSPC          ;Skip over spaces
                                ;
                                ;  Find devices for deallocation.
                                ;
001320  111300                  2$:    MOVB    (R3),R0         ;Get 1st char of operand
001322  001473                        BEQ     10$             ;No devices specified, deallocate all
001324  120027  000057          3$:    CMPB    R0,#'/          ;Was a switch specified?
001330  001012                        BNE     4$              ;Br if not
001332  005203                        INC     R3              ;Skip over slash
001334  004767  000000G               CALL    SKPSPC          ;Skip any spaces
001340  122327  000101                CMPB    (R3)+,#'A       ;"ALL"?
001344  001465                        BEQ     12$             ;Br if yes
001346                                FABORT  #INVOPT         ;Invalid option
                                ;
                                ;  Deallocate a list of devices
                                ;  Accrue the next device name
                                ;
001356  005267  176420          4$:    INC     ALC1DV          ;Count deallocated devices
001362  004767  000000G               CALL    SKPSPC          ;Skip over any spaces
001366  004767  000000G               CALL    GTRD50          ;Accrue the device name
001372  122327  000072                CMPB    (R3)+,#':       ;Was colon specified following name?
001376  001401                        BEQ     5$              ;Br if yes
001400  005303                        DEC     R3              ;Backup pointer
001402  004767  000000G         5$:    CALL    SKPDLM          ;Skip valid delimiters (blank(s), comma)
001406  103435                        BCS     7$              ;Br if invalid character
                                ;
                                ;  Deallocate the device specified.
                                ;
001410  016702  000000G               MOV     R50BUF,R2       ;Get the name of the device
001414  010267  000000G               MOV     R2,ALCDEV       ;Set name of device being allocated
001420  012700  000000G               MOV     #DLCEMT,R0      ;Point to EMT argument block
001424  104375                        EMT     375             ;Do the deallocation
001426  103334                        BCC     2$              ;Br if ok
001430  010005                        MOV     R0,R5           ;Save job # if owned by someone else
001432  113700  000000G               MOVB    @#ERRLOC,R0     ;Get error code
001436  120227  000001                CMPB    R2,#1           ;Device allocated by someone else?
001442  001014                        BNE     6$              ;Br if not
001444                                FERR    #EM$DAA         ;Device allocated by another job
001460  004767  000000G               CALL    PRTDEC          ;Print the number of the job that has the dev
001464                                .PRINT  #CRLF           ;End of line
001472  000712                        BR      2$
001474  120227  000002          6$:    CMPB    R2,#2           ;Invalid device being allocated?
001500  001307                        BNE     2$              ;Br if not
001502                          7$:    FABORT  #EM$IAD         ;Invalid device for allocation
```

```
                                        ;
                                        ;  Deallocate all devices allocated by this user
                                        ;
001512  005767  176264          10$:    TST     ALC1DV          ;Were any devices specified?
001516  001006                          BNE     20$             ;Yes, end of command
001520  005067  000000G         12$:    CLR     ALCDEV          ;Say to deallocate all devices
001524  012700  000000G                 MOV     #DLCEMT,R0      ;Point to EMT argument block
001530  104375                          EMT     375             ;Do the deallocation
001532  000400                          BR      20$
                                        ;
                                        ;  Finished
                                        ;
001534  000167  000000G         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.
                                ;
001540  004767  000000G       CMDREM: CALL    CVTTAB          ;Convert tab and FF chars to spaces
                                ;
                                ;  Quit if we have reached the end of the command line
                                ;
001544  004767  000000G       1$:     CALL    SKPSPC          ;Skip past leading spaces
001550  105713                        TSTB    (R3)            ;Are we at the end of the command?
001552  001436                        BEQ     9$              ;Br if yes
                                ;
                                ;  Accrue the name of a device or region
                                ;
001554  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.
                                ;
001560  005767  000002G               TST     R50BUF+2        ;Is name longer than 3 characters?
001564  001012                        BNE     2$              ;Br if yes
001566  121327  000072                CMPB    (R3),#':        ;Is name terminated with colon?
001572  001002                        BNE     3$              ;Br if not
001574  105723                        TSTB    (R3)+           ;Skip over colon
001576  000407                        BR      5$              ;Must have been a device name
001600  016705  000000G       3$:     MOV     R50BUF,R5       ;Get 1st 3 chars of name
001604  004767  000000G               CALL    CHKDEV          ;See if this is the name of a device
001610  103002                        BCC     5$              ;Br if device name
                                ;
                                ;  This is a request to remove a named PLAS region.
                                ;
001612  004767  000036        2$:     CALL    REMRGN          ;Remove a region
                                ;
                                ;  Skip over any separating comma
                                ;
001616  122327  000054        5$:     CMPB    (R3)+,#',       ;Is there a separating comma?
001622  001401                        BEQ     4$              ;Br if yes
001624  005303                        DEC     R3              ;Point back to skipped char
001626  105713                4$:     TSTB    (R3)            ;Reached end of command?
001630  001407                        BEQ     9$              ;Br if yes
001632  121327  000040                CMPB    (R3),#' '       ;Space separator?
001636  001742                        BEQ     1$              ;Br if yes
001640                                FABORT  #EM$CSE         ;Command syntax error
                                ;
                                ;  Finished
                                ;
001650  000167  000000G       9$:     JMP     RDCMD
```

```
                                ;-------------------------------------------------------------------------------
                                ;  Eliminate a named PLAS region.
                                ;
                                ;  Inputs:
                                ;   R50BUF = Rad50 name of the region.
                                ;
001654  010246                  REMRGN: MOV     R2,-(SP)
                                ;
                                ;  First attempt to attach to a private region with that name.
                                ;
001656  012702  000020'                 MOV     #REMRDB,R2      ;Point to region definition block
001662  005062  000000G                 CLR     R.GSIZ(R2)      ;Clear size word
001666  012762  000000C 000000G         MOV     #<RS.GBL!RS.PVT>,R.GSTS(R2) ;Function = Attach private region
001674  016762  000000G 000000G         MOV     R50BUF,R.NAME(R2) ;Set name of region
001702  016762  000002G 000002G         MOV     R50BUF+2,R.NAME+2(R2)
001710                                  .CRRG   #XAREA,R2       ;Attempt to attach to region
001726  032762  000000G 000000G         BIT     #RS.CRR,R.GSTS(R2) ;Were we able to attach?
001734  001025                          BNE     1$              ;Br if yes
                                ;
                                ;  Unable to attach to private region.
                                ;  Try to attach to public region.
                                ;
001736  012762  000000G 000000G         MOV     #RS.GBL,R.GSTS(R2) ;Function = Attach global region
001744                                  .CRRG   #XAREA,R2       ;Attempt to attach to it
001762  032762  000000G 000000G         BIT     #RS.CRR,R.GSTS(R2) ;Was attachment successful?
001770  001007                          BNE     1$              ;Br if yes
                                ;
                                ;  Error -- We are unable to attach to region
                                ;
001772                                  FERR    #EM$UAR         ;Unable to attach to region
002006  000421                          BR      9$
                                ;
                                ;  We successfully attached to the region.
                                ;  Try to eliminate the region.
                                ;
002010  012762  000000G 000000G 1$:     MOV     #RS.EGR,R.GSTS(R2) ;Function = Eliminate region
002016                                  .ELRG   #XAREA,R2       ;Try to eliminate the region
002034  103006                          BCC     9$              ;Br if successful
                                ;
                                ;  Error -- Unable to eliminate the region
                                ;
002036                                  FERR    #EM$UER         ;Unable to eliminate region
                                ;
                                ;  Finished
                                ;
002052  012602                  9$:     MOV     (SP)+,R2
002054  000207                          RETURN
```

```
                                        .SBTTL  INSTALL Command
                                ;-----------------------------------------------------------------
                                ;  Process the INSTALL command.
                                ;
002056  004767  0000000G        CMDINS: CALL    CVTTAB          ;Convert tabs to spaces
002062  004767  0000000G                CALL    CVTUC           ;Convert to upper case
                                ;
                                ;  Accrue first keyword and see if it is ADD or DELETE
                                ;
002066  010302                          MOV     R3,R2           ;Save pointer to start of keyword
002070  004767  0000000G                CALL    GTRD50          ;Accrue 1st keyword
002074  016700  0000000G                MOV     R50BUF,R0       ;Get 1st 3 chars of keyword
002100  020067  175726                  CMP     R0,R50ADD       ;Is keyword ADD?
002104  001421                          BEQ     INSADD          ;Br if yes
002106  020067  175722                  CMP     R0,R50DEL       ;Is keyword DELETE?
002112  001473                          BEQ     INSDEL          ;Br if yes
002114  020067  175720                  CMP     R0,R50REM       ;Is keyword REMOVE?
002120  001470                          BEQ     INSDEL          ;Br if yes
                                ;
                                ;  Keyword is not ADD or DELETE.
                                ;  Error if 1st keyword is longer than 2 characters.
                                ;
002122  010300                          MOV     R3,R0           ;Get pointer past end of keyword
002124  160200                          SUB     R2,R0           ;Determine length of 1st keyword
002126  020027  000002                  CMP     R0,#2           ;1st keyword longer than 2 chars?
002132  101404                          BLOS    1$              ;Br if not
002134                                  FABORT  #ILLCMD         ;Invalid command
                                ;
                                ;  This must be a device handler name -- Ignore this command
                                ;
002144  000167  0000000G        1$:     JMP     RDCMD           ;Finished installing device handler
```

```
                                ;-------------------------------------------------------------------------
                                ;  We are adding an image
                                ;
002150  004767  0000006G        INSADD: CALL    CKSYPV          ;Require SYSPRV privilege
002154  005067  175662                  CLR     IATTRB          ;Clear all attribute flags
002160  004767  0000006G                CALL    CLRPRV          ;Reset privilege flag cells
002164  004767  000546                  CALL    INSOPT          ;Process any options that follow ADD
                                ;
                                ;  Get the name of the image and see if it is currently in table
                                ;
002170  004767  000144                  CALL    INSNAM          ;Get name of image and look it up
002174  103002                        BCC     1$              ;Br if name already in table
                                ;
                                ;  Name is not currently in table.
                                ;  Try to find a free slot in the table.
                                ;
002176  004767  000450                  CALL    INSFRE          ;Find free table entry
                                ;
                                ;  Set up program name for free entry
                                ;
002202  012702  000000C         1$:     MOV     #II$NAM+IIBUF,R2;Point to target buffer
002206  012704  0000006G                MOV     #FILNAM,R4      ;Point to name
002212  012700  000004                  MOV     #4,R0           ;Move 4 words
002216  012422                  2$:     MOV     (R4)+,(R2)+     ;Move name to buffer
002220  077002                          SOB     R0,2$
                                ;
                                ;  At this point, R5 points to buffer where install entry is to be made
                                ;  Process any options that follow the file name
                                ;
002222  004767  000510                  CALL    INSOPT          ;Process any options that follow name
                                ;
                                ;  Move attribute and privilege flags into install entry
                                ;
002226  010546                          MOV     R5,-(SP)        ;Save address of entry in install table
002230  016767  175606  000001C         MOV     IATTRB,II$FLG+IIBUF ;Set attribute flags for program
002236  012702  0000006G                MOV     #PFS0,R2        ;Point to accrued privilege flags
002242  012703  000000C                 MOV     #II$PRV+IIBUF,R3;Install entry flags
002246  012704  0000006G                MOV     #PFC0,R4        ;Flags to clear when run
002252  012705  000000C                 MOV     #II$NPV+IIBUF,R5;Install entry
002256  012700  0000006G                MOV     #PVNPW,R0       ;Get # words to move
002262  012223                  3$:     MOV     (R2)+,(R3)+     ;Move flags to be set
002264  012425                          MOV     (R4)+,(R5)+     ;Move flags to be cleared
002266  077003                          SOB     R0,3$
002270  012605                          MOV     (SP)+,R5        ;Recover address of entry in install table
                                ;
                                ;  Add the entry to the install table
                                ;
002272  004767  000424                  CALL    INSPUT          ;Add entry to table
                                ;
                                ;  Finished
                                ;
002276  000167  0000006G                JMP     RDCMD
```

```
      1                                 ;
      2                                 ;  Process INSTALL DELETE command
      3                                 ;
      4 002302  004767  000000G        INSDEL: CALL    CKSYPV          ;Require SYSPRV privilege
      5                                 ;
      6                                 ;  See if we can find program in install table
      7                                 ;
      8 002306  004767  000026               CALL    INSNAM          ;Accrue file spec and try to find in table
      9 002312  103004                       BCC     1$              ;Br if found file name in table
     10 002314                               FABORT  #EM$FNI         ;File not installed
     11                                 ;
     12                                 ;  Found entry, delete it
     13                                 ;
     14 002324  005067  000001C        1$:   CLR     II$NAM+IIBUF    ;Say image is not installed
     15 002330  004767  000366               CALL    INSPUT          ;Write entry back to table
     16                                 ;
     17                                 ;  Finished
     18                                 ;
     19 002334  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.
                                ;
002340  010246                  INSNAM: MOV     R2,-(SP)
002342  010446                          MOV     R4,-(SP)
                                ;
                                ;  Determine if wildcard ("*") specified as device name
                                ;
002344  005002                          CLR     R2              ;Assume device name is not wild
002346  004767  0000006                 CALL    SKPSPC          ;Skip up to start of file spec
002352  121327  000052                  CMPB    (R3),#'*        ;Wildcard as device name?
002356  001013                          BNE     11$             ;Br if not
002360  126327  000001  000072          CMPB    1(R3),#':       ;Is this the device name?
002366  001007                          BNE     11$             ;Br if not
002370  012702  132500                  MOV     #WLDNAM,R2      ;Remember device name is wild
002374  112713  000040                  MOVB    #' ,(R3)        ;Blank out the device name
002400  112763  000040  000001          MOVB    #' ,1(R3)
                                ;
                                ;  Accrue the file spec
                                ;
002406  012704  0000006         11$:    MOV     #R50SAV,R4      ;Get default file extension (SAV)
002412  005005                          CLR     R5              ;Say this is an input file
002414  004767  0000006                 CALL    ACRFIL          ;Accrue the file spec
002420  103510                          BCS     10$             ;Br if invalid file spec
002422  005767  0000026                 TST     FILNAM+2        ;Was a file name specified?
002426  001505                          BEQ     10$             ;Br if not
002430  005702                          TST     R2              ;Is device name wild?
002432  001403                          BEQ     12$             ;Br if not
002434  010267  0000006                 MOV     R2,FILNAM       ;Set wildcard as file name
002440  000431                          BR      5$
                                ;
                                ;  Translate logical file device to physical device with unit #
                                ;
002442  012705  0000006         12$:    MOV     #FILNAM,R5      ;Point to file spec
002446  004767  0000006                 CALL    LOGASN          ;Do logical assignment
002452  010346                          MOV     R3,-(SP)        ;SAVE POINTER
002454  010503                          MOV     R5,R3           ;GET COPY OF DEVICE POINTER
002456  004767  0000006                 CALL    FORCE0          ;MAKE BLANK UNIT = 0
002462  012603                          MOV     (SP)+,R3        ;RESTORE POINTER
                                ;
                                ;  Error if logical disk (LD) specified for device
                                ;
002464  016705  0000006                 MOV     FILNAM,R5       ;Get the device name
002470  004767  0000006                 CALL    CHKDEV          ;Convert to device index number
002474  103407                          BCS     6$              ;Br if invalid device
002476  020467  0000006                 CMP     R4,LDDEVX       ;Is this a logical disk?
002502  001010                          BNE     5$              ;Br if not
002504                                  FABORT  #EM$LDI         ;Can't have installed program on LD
```

```
     58 002514                                6$:     FABORT  #EM$IAD          ;Invalid device
     59                                       ;
     60                                       ;  Try to find existing entry for file in install table
     61                                       ;
     62 002524  016705  000000G               5$:     MOV     INSTBL,R5        ;Point to 1st entry in table
     63 002530  010567  000000G               1$:     MOV     R5,INGADR        ;Set address in EMT
     64 002534  012700  000000G                       MOV     #INGEMT,R0       ;Point to get EMT
     65 002540  104375                               EMT     375              ;Get next install table entry
     66 002542  005760  000000C                       TST     IIBUF+II$NAM(R0);Is this entry empty?
     67 002546  001422                               BEQ     2$               ;Br if yes
     68 002550  012700  000006                       MOV     #6,R0            ;Get index to last word of name
     69 002554  026027  000000G 132500        3$:     CMP     FILNAM(R0),#WLDNAM ;Wildcard?
     70 002562  001410                               BEQ     13$              ;Br if yes
     71 002564  026027  000000C 132500               CMP     IIBUF+II$NAM(R0),#WLDNAM ;Wildcard?
     72 002572  001404                               BEQ     13$              ;Br if yes
     73 002574  026060  000000G 000000C              CMP     FILNAM(R0),IIBUF+II$NAM(R0) ;Compare names
     74 002602  001004                               BNE     2$               ;Br if this is not right entry
     75 002604  162700  000002              13$:     SUB     #2,R0            ;More words to check?
     76 002610  002361                               BGE     3$               ;Br if yes
     77 002612  000407                               BR      4$               ;Found entry
     78 002614  062705  000000G               2$:     ADD     #II$$SZ,R5       ;Point to next entry in table
     79 002620  020567  000000G                       CMP     R5,INSTBN        ;Reached end of table?
     80 002624  103741                               BLO     1$               ;Loop if not
     81                                       ;
     82                                       ;  Cannot find entry in table
     83                                       ;
     84 002626  000261                               SEC                      ;Signal failure on return
     85 002630  000401                               BR      9$
     86                                       ;
     87                                       ;  We found entry for this name
     88                                       ;
     89 002632  000241                        4$:    CLC                      ;Signal success on return
     90                                       ;
     91                                       ;  Finished
     92                                       ;
     93 002634  012604                        9$:    MOV     (SP)+,R4
     94 002636  012602                               MOV     (SP)+,R2
     95 002640  000207                               RETURN
     96                                       ;
     97                                       ;  Invalid file spec
     98                                       ;
     99 002642                                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.
                                ;
002652  016705  0000000G      INSFRE: MOV     INSTBL,R5       ;Point to 1st install table entry
002656  010567  0000000G      1$:     MOV     R5,INGADR       ;Set address for EMT
002662  012700  0000000G              MOV     #INGEMT,R0      ;EMT to get assign table entry
002666  104375                      EMT     375             ;Get next entry
002670  005767  000001C             TST     II$NAM+IIBUF    ;Is this entry free?
002674  001411                      BEQ     2$              ;Br if found free entry
002676  062705  0000000G            ADD     #II$$SZ,R5      ;Point to next entry
002702  020567  0000000G            CMP     R5,INSTBN       ;Reached end of table?
002706  103763                      BLO     1$              ;Loop if not
                                ;
                                ;  No free table entries
                                ;
002710                              FABORT  #EM$ITF         ;Install table full
                                ;
                                ;  Found free entry
                                ;
002720  000207              2$:     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 002722  010567  000000G         INSPUT: MOV     R5,INPADR       ;Set address for EMT
      8 002726  012700  000000G                 MOV     #INPEMT,R0      ;Point to EMT
      9 002732  104375                          EMT     375             ;Store install table entry
     10 002734  000207                          RETURN
```

```
                                ;-----------------------------------------------------------------------
                                ;  Process command options specified with an INSTALL command.
                                ;
                                ;  Inputs:
                                ;   R3 = Pointer to option list.
                                ;
                                ;  Outputs:
                                ;   R3 = Points past end of option list
                                ;
002736  010446                  INSOPT: MOV     R4,-(SP)
                                ;
                                ;  Process list of options
                                ;
002740  012704  000000'                 MOV     #INSHD,R4       ;Point to table of options
002744  004767  000000G                 CALL    OPTLST          ;Process an option list
                                ;
                                ;  Finished
                                ;
002750  012604                          MOV     (SP)+,R4
002752  000207                          RETURN
                                ;-----------------------------------------------------------------------
                                ;  Subroutine called to set a specific install attribute flag
                                ;
                                ;  Inputs:
                                ;   R4 = Pointer to option table entry.
                                ;
                                ;  Outputs:
                                ;   IATTRB = Combined attribute flags
                                ;
002754  051467  175062          INSATR: BIS     (R4),IATTRB     ;Set attribute flag
002760  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
                                ;
002762  004767  000000G       CMDMNT: CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACES
                                ;
                                ;  Determine if a command switch was specified
                                ;
002766  105067  000000G               CLRB    NOFLAG          ;ASSUME /NOWRITE NOT WANTED
002772  122327  000040        1$:     CMPB    (R3)+,#' '      ;SKIP ANY LEADING SPACES
002776  001775                        BEQ     1$
003000  124327  000057                CMPB    -(R3),#'/       ;WAS A SWITCH SPECIFIED?
003004  001014                        BNE     3$              ;BR IF NOT
003006  005203                        INC     R3              ;SKIP PAST /
003010  012704  000160'              MOV     #MTOPHD,R4      ;POINT TO TABLE OF SWITCHES
003014  004767  000000G               CALL    SEARCH          ;LOOK UP THE SWITCH
003020  103004                        BCC     2$              ;BR IF FOUND
003022                                FABORT  #INVOPT         ;INVALID SWITCH
003032  111467  000000G       2$:     MOVB    (R4),NOFLAG     ;REMEMBER IF NOWRITE SEEN
                                ;
                                ;  Get device name
                                ;
003036  105713                3$:     TSTB    (R3)            ;IS THERE A DEVICE NAME?
003040  001004                        BNE     12$             ;BR IF YES
003042                                FABORT  #ILLCMD         ;INVALID COMMAND
003052  004767  000000G       12$:    CALL    GTRD50          ;GET DEVICE NAME
003056  016702  000000G               MOV     R50BUF,R2       ;PICK UP DEVICE NAME
003062  020267  000000G               CMP     R2,R50LD        ;IS NAME "LD"?
003066  001002                        BNE     15$             ;BR IF NOT
003070  016702  000000G               MOV     R50LD0,R2       ;TRANSLATE LD TO LD0
003074  010267  000000G       15$:    MOV     R2,MNTDEV       ;SAVE DEVICE NAME
003100  122327  000072                CMPB    (R3)+,#':       ;WAS DEVICE NAME TERMINATED WITH COLON?
003104  001401                        BEQ     4$              ;BR IF YES
003106  005303                        DEC     R3              ;BACKUP POINTER IF NOT
                                ;
                                ;  See if device we are mounting is allocated to another user
                                ;
003110  016767  000000G 000000G 4$:     MOV     MNTDEV,ALCDEV   ;Set device name for allocation EMT
003116  012700  000000G               MOV     #TALEMT,R0      ;Point to EMT argument block
003122  104375                        EMT     375             ;Check for allocation conflict
003124  103022                        BCC     5$              ;Br if no allocation failure
003126  123727  000000G 000001        CMPB    @#ERRLOC,#1     ;Device allocated to another user?
003134  001016                        BNE     5$              ;Br if not
003136  010005                        MOV     R0,R5           ;Save # of job that owns the device
003140                                FERR    #EM$DAA         ;Device allocated to job
003154  004767  000000G               CALL    PRTDEC          ;Print # of job that owns device
003160                                .PRINT  #CRLF           ;End print line
003166  000167  000000G               JMP     RDCMD           ;Abort command
                                ;
                                ;  See if there is a file name specified.
                                ;
003172  004767  000000G       5$:     CALL    SKPSPC          ;Skip any spaces
003176  105713                        TSTB    (R3)            ;Was a file name specified?
003200  001002                        BNE     MNTLD           ;BR IF YES
```

```
     58 003202  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.
                                ;
003206  105767  000000G        MNTLD:  TSTB    VLDSYS          ;WAS LD SUPPORT GENNED IN?
003212  001004                        BNE     18$             ;BR IF YES
003214                          .       FABORT  #EM$NLD         ;LD SUPPORT NOT GENNED IN
                                ;
                                ;  Make sure the logical disk name is ok.
                                ;
003224  010200                  18$:    MOV     R2,R0           ;GET LD DEVICE NAME
003226  004767  000412                  CALL    LDDEVN          ;GET LD DEVICE INDEX NUMBER
003232  103004                        BCC     6$              ;BR IF OK
003234                          .       FABORT  #INVLDN         ;ERROR IF NAME IS NOT LD
003244  010002                  6$:     MOV     R0,R2           ;CARRY INDEX IN R2
                                ;
                                ;  Close the log file if it is on the logical disk
                                ;
003246  016705  000000G                 MOV     MNTDEV,R5       ;GET NAME OF LOGICAL DEVICE
003252  004767  000000G                 CALL    LOGCHK          ;SEE IF WE NEED TO CLOSE THE LOG FILE
                                ;
                                ;  Tell system to stop doing directory caching for the device
                                ;
003256  112767  000001  000000G       MOVB    #1,SERFLG       ;DO .SERR TO AVOID ABORT FOR ILLEGAL DEVICE
003264  012700  000000G               MOV     #DMTARG,R0      ;POINT TO EMT ARGUMENT BLOCK
003270  104375                        EMT     375             ;TELL SYSTEM TO STOP DOING CACHING
003272  105067  000000G               CLRB    SERFLG          ;DO .HERR
                                ;
                                ;  Accrue the file name
                                ;
003276  012704  000000G               MOV     #R50DSK,R4      ;SET DEFAULT FILE EXTENSION
003302  005005                        CLR     R5              ;SAY THIS IS AN INPUT FILE
003304  004767  000000G               CALL    ACRFIL          ;ACCRUE THE FILE SPEC
003310  103004                        BCC     13$             ;BR IF OK
003312                          5$:     FABORT  #BDFNAM         ;INVALID FILE SPEC
                                ;
                                ;  Translate logical file device name to physical device
                                ;
003322  012705  000000G         13$:    MOV     #FILNAM,R5      ;POINT TO FILE SPEC AREA
003326  004767  000000G                 CALL    LOGASN          ;PERFORM ANY LOGICAL DEVICE ASSIGNMENT
                                ;
                                ;  Make sure file is not on same LD as is being mounted
                                ;
003332  016700  000000G               MOV     FILNAM,R0       ;GET DEVICE THAT FILE IS ON
003336  004767  000302                CALL    LDDEVN          ;SEE IF FILE IS ON A LD DEVICE
003342  103406                        BCS     17$             ;BR IF NOT ON LD DEVICE
003344  020002                        CMP     R0,R2           ;IS IT ON LOWER # LD?
003346  103404                        BLO     17$             ;BR IF YES
003350                          .       FABORT  #INVLDM         ;INVALID LD ORDER
                                ;
                                ;  Make sure the file name is not null
                                ;
003360  005767  000002G         17$:    TST     FILNAM+2        ;Is file name null?
003364  001752                        BEQ     5$              ;Br if yes -- Error
                                ;
                                ;  Move file name into logical disk name table
```

```
                        ;
003366  010205          MOV     R2,R5           ;GET LOGICAL DISK UNIT #
003370  006305          ASL     R5              ;*4 WORDS PER ENTRY
003372  006305          ASL     R5
003374  062705  000000G ADD     #LDNAME,R5      ;POINT TO ENTRY IN LOGICAL DISK TABLE
003400  012704  000000G MOV     #FILNAM,R4      ;POINT TO FILE NAME BUFFER
003404  012425          MOV     (R4)+,(R5)+     ;MOVE NAME TO TABLE
003406  012425          MOV     (R4)+,(R5)+
003410  012425          MOV     (R4)+,(R5)+
003412  011415          MOV     (R4),(R5)
                        ;
                        ;  Set up flags for logical disk
                        ;
003414  005004          CLR     R4
003416  105767  000000G TSTB    NOFLAG          ;WAS NOWRITE SPECIFIED?
003422  001402          BEQ     9$              ;BR IF NOT
003424  052704  000000G BIS     #LD$RON,R4      ;SET READ-ONLY FLAG
003430  010462  000000G 9$:     MOV     R4,LDFLAG(R2)
                        ;
                        ;  Lookup file and set up information about position of logical
                        ;  disk on physical disk.
                        ;
003434  004767  000000G CALL    LDMNT           ;SET UP INFO ABOUT FILE FOR LOGICAL DISK
003440  103004          BCC     10$             ;BR IF FOUND FILE
003442                  FABORT  #CSIMS4         ;CAN'T FIND FILE
                        ;
                        ;  Protect the file
                        ;
003452  010205          10$:    MOV     R2,R5           ;GET LOGICAL DISK UNIT #
003454  006305          ASL     R5              ;* 4 WORDS PER ENTRY
003456  006305          ASL     R5
003460  062705  000000G ADD     #LDNAME,R5      ;POINT TO ENTRY WITH FILE NAME
003464  112767  000001  000000G MOVB    #1,SERFLG       ;DO .SERR TO AVOID ABORTS ON ERRORS
003472                  .FPROT  #XAREA,#1,R5,#1
003516  105067  000000G CLRB    SERFLG          ;DO .HERR
                        ;
                        ;  See if a logical device name was specified following the file name
                        ;
003522  122327  000040  16$:    CMPB    (R3)+,#' ;SKIP BLANKS
003526  001775          BEQ     16$
003530  105743          TSTB    -(R3)           ;WAS A LOGICAL DEVICE NAME SPECIFIED?
003532  001410          BEQ     MNTCSH          ;BR IF NOT
                        ;
                        ;  Assign logical name to logical disk
                        ;
003534  004767  000000G CALL    GTRD50          ;ACCRUE LOGICAL DEVICE NAME
003540  016705  000000G MOV     R50BUF,R5       ;GET LOGICAL DEVICE NAME
003544  016700  000000G MOV     MNTDEV,R0       ;GET PHYSICAL DEVICE NAME
003550  004767  000000G CALL    DOASGN          ;DO THE ASSIGNMENT
                        ;
                        ;  Initiate directory caching for the device whose name is in MNTDEV
                        ;
003554  012700  000000G MNTCSH: MOV     #MNTARG,R0      ;POINT TO MOUNT ARGUMENT BLOCK
003560  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 003564  004767  000000G                 CALL    LDCLEN          ;DO SET LD CLEAN
    117                                 ;
    118                                 ;  Finished with MOUNT command
    119                                 ;
    120 003570  000167  000000G                 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.
                                ;
003574  104375                 DOMNT:  EMT     375             ;DO THE EMT (MOUNT OR DISMOUNT)
003576  103021                         BCC     9$              ;BR IF NO ERROR
                                ;
                                ;  Error on mount or dismount emt.
                                ;
003600  123727  0000006 000001         CMPB    @#ERRLOC,#1     ;MOUNT TABLE OVERFLOW?
003606  001007                         BNE     2$              ;BR IF NOT
003610                                 FWARN   #MNTFUL         ;MOUNT TABLE OVERFLOW
003624  000406                         BR      9$
003626                          2$:    FERR    #BDFNAM         ;INVALID DEVICE NAME
                                ;
                                ;  Finished
                                ;
003642  000207                  9$:    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)
                                ;
003644  020067  0000000 LDDEVN: CMP     R0,R50LD        ;Is name "LD"?
003650  001002                  BNE     1$              ;Br if not
003652  016700  0000000         MOV     R50LD0,R0       ;Replace "LD" by "LD0"
003656  020067  0000000 1$:     CMP     R0,R50LD0       ;Is name in the range LD0 to LD7?
003662  103410                  BLO     2$              ;Br if not
003664  020067  0000000         CMP     R0,R50LD7
003670  101005                  BHI     2$
003672  166700  0000000         SUB     R50LD0,R0       ;Get unit number only
003676  006300                  ASL     R0              ;Convert to device index number
003700  000241                  CLC                     ;Signal success on return
003702  000401                  BR      3$
003704  000261          2$:     SEC                     ;Signal failure on return
003706  000207          3$:     RETURN
```

```
                                        .SBTTL  DISMOUNT command
                                ;-----------------------------------------------------------------------------
                                ;  Dismount a file structure.
                                ;
003710  004767  000000G       CMDDMT: CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACES
                                ;
                                ;  See if any qualifier were specified following command keyword
                                ;
003714  105067  174124              CLRB    DMTNLG          ;Clear flag set if /NOLOG specified
003720  004767  000000G              CALL    SKPSPC          ;Skip over any spaces
003724  121327  000057              CMPB    (R3),#'/        ;Any qualifiers here?
003730  001004                       BNE     5$              ;Br if not
                                ;
                                ;  Process qualifier following command keyword
                                ;
003732  012704  000174'              MOV     #DMTQHD,R4      ;Point to parsing table
003736  004767  000000G              CALL    OPTLST          ;Process any qualifiers
                                ;
                                ;  Skip leading spaces and make sure a device name was specified
                                ;
003742  004767  000000G       5$:    CALL    SKPSPC          ;SKIP LEADING SPACES
003746  105713                       TSTB    (R3)            ;WAS A DEVICE NAME SPECIFIED
003750  001004                       BNE     6$              ;BR IF YES
003752                         7$:    FABORT  #ILLCMD         ;INVALID COMMAND IF NO DEVICE NAME
                                ;
                                ;  Get name of device that is being dismounted
                                ;
003762  004767  000000G       6$:    CALL    GTRD50          ;ACCRUE THE DEVICE NAME
003766  016700  000000G              MOV     R50BUF,R0       ;GET DEVICE NAME
003772  001767                       BEQ     7$              ;BR IF NO DEVICE NAME
003774  004767  000000G       2$:    CALL    ASNSRC          ;SEE IF IT IS A LOGICAL NAME
004000  103403                       BCS     1$              ;BR IF NOT
004002  016200  000000G              MOV     AT$DEV(R2),R0   ;GET ASSIGNED NAME
004006  000772                       BR      2$              ;ALLOW INDIRECT ASSIGNS
004010  020067  000000G       1$:    CMP     R0,R50LD        ;IS NAME "LD"?
004014  001002                       BNE     4$              ;BR IF NOT
004016  016700  000000G              MOV     R50LD0,R0       ;TRANSLATE LD TO LD0
004022  010067  000000G       4$:    MOV     R0,MNTDEV       ;SAVE NAME OF DEVICE BEING DISMOUNTED
                                ;
                                ;  Close the log file if it is on the device being dismounted
                                ;
004026  016705  000000G              MOV     MNTDEV,R5       ;GET NAME OF DEVICE BEING DISMOUNTED
004032  004767  000000G              CALL    LOGCHK          ;SEE IF LOG FILE IS ON THAT DEVICE
                                ;
                                ;  Print information message if device is still mounted by other jobs
                                ;
004036  105767  174002              TSTB    DMTNLG          ;Was /NOLOG qualifier specified?
004042  001013                       BNE     3$              ;Br if yes -- Suppress information message
004044  016705  000000G              MOV     MNTDEV,R5       ;GET NAME OF DEVICE BEING DISMOUNTED
004050  004767  000000G              CALL    CHKMNT          ;IS THIS DEVICE MOUNTED?
004054  103406                       BCS     3$              ;BR IF NOT
004056  004767  000000G              CALL    CHKMTX          ;IS DEVICE MOUNTED BY ANYONE OTHER THAN US?
004062  103003                       BCC     3$              ;BR IF NOT
004064                               .PRINT  #INFOMT         ;STILL MOUNTED BY OTHER USERS
                                ;
                                ;  Tell system to stop doing directory caching for the device
                                ;
```

```
     58 004072  112767  000001  000000G 3$:     MOVB    #1,SERFLG       ;DO .SERR TO AVOID ABORT FOR ILLEGAL DEVICE
     59 004100  012700  000000G                 MOV     #DMTARG,R0      ;POINT TO EMT ARGUMENT BLOCK
     60 004104  104375                          EMT     375             ;TELL SYSTEM TO STOP DOING CACHING
     61 004106  105067  000000G                 CLRB    SERFLG          ;DO .HERR
     62                                 ;
     63                                 ;  See if a logical disk is being dismounted
     64                                 ;
     65 004112  016702  000000G                 MOV     MNTDEV,R2       ;GET DEVICE NAME
     66 004116  020267  000000G                 CMP     R2,R50LD0       ;IS THIS A LOGICAL DISK?
     67 004122  103420                          BLO     9$              ;BR IF NOT
     68 004124  020267  000000G                 CMP     R2,R50LD7
     69 004130  101015                          BHI     9$
     70 004132  166702  000000G                 SUB     R50LD0,R2       ;GET UNIT #
     71 004136  006302                          ASL     R2              ;CONVERT TO WORD INDEX
     72 004140  005062  000000G                 CLR     LDPDEV(R2)      ;NO PHYSICAL DEVICE ASSIGNMENT
     73 004144  006302                          ASL     R2              ;GET INDEX INTO LDNAME TABLE
     74 004146  006302                          ASL     R2
     75 004150  005062  000000G                 CLR     LDNAME(R2)      ;NO FILE NAME
     76 004154  016700  000000G                 MOV     MNTDEV,R0       ;GET NAME OF LOGICAL DISK
     77 004160  004767  000000G                 CALL    DEADEV          ;DEASSIGN ANY LOGICAL NAMES
     78                                 ;
     79                                 ;  Finished
     80                                 ;
     81 004164  004767  000000G         9$:     CALL    LDCLEN          ;RESET INFORMATION ABOUT LOGICAL DISKS
     82 004170  000167  000000G                 JMP     RDCMD           ;FINISHED WITH COMMAND
     83                                 ;
     84                                 ;  Process /NOLOG qualifier for DISMOUNT command
     85                                 ;
     86 004174  105267  173644          DMTNLQ: INCB    DMTNLG          ;Remember qualifier specified
     87 004200  000207                  DMTLGQ: RETURN
```

```
                                        .SBTTL  INITIALIZE command
                                        .SBTTL  SQUEEZE command
                                        .SBTTL  FORMAT command
                                ;-----------------------------------------------------------------------
                                ;  Disallow INITIALIZE and SQUEEZE commands if any other users have
                                ;  the specified device mounted.
                                ;
004202                          CMDINI:
004202                          CMDSQZ:
004202                          CMDFMT:
                                ;
                                ;  See if user is authorized to use these commands
                                ;
004202  032767  000000C 000000G         BIT     #PO$NFW!PO$BYP,PRIVCO   ;Authorized for these commands?
004210  001004                          BNE     13$             ;Br if yes
004212                                  FABORT  #EM$NFW         ;Not authorized for this command
                                ;
                                ;  Skip over any options specified with command keyword
                                ;
004222  004767  000000G         13$:    CALL    CVTTAB          ;Convert tabs to spaces
004226  121327  000057                  CMPB    (R3),#'/        ;Were any keyword options specified?
004232  001005                          BNE     10$             ;Br if not
004234  112300                  1$:     MOVB    (R3)+,R0        ;Get next character from command
004236  001407                          BEQ     15$             ;Br if no device was specified
004240  120027  000040                  CMPB    R0,#40          ;Is this a space?
004244  001373                          BNE     1$              ;Keep looking if not
                                ;
                                ;  Accrue the device name
                                ;
004246  004767  000000G         10$:    CALL    SKPSPC          ;Skip over any spaces
004252  105713                          TSTB    (R3)            ;Was a device name specified?
004254  001004                          BNE     14$             ;Br if yes
004256                          15$:    FABORT  #EM$DNR         ;Device name is required
004266  004767  000000G         14$:    CALL    GTRD50          ;Accrue the device name
004272  016705  000000G                 MOV     R50BUF,R5
004276  010567  000000G                 MOV     R5,MNTDEV       ;Set name for mount/dismount
004302  010567  173472                  MOV     R5,SQZDEV       ;Save name of device being squeezed
004306  010567  000000G                 MOV     R5,ALCDEV       ;Set name for test-allocation EMT
                                ;
                                ;  Disallow SQUEEZE/INITIALIZE/FORMAT of any disk with a system file
                                ;
004312  004767  000232                  CALL    CKSYDV          ;See if device has any system files
004316  103004                          BCC     11$             ;Br if not
004320                                  FABORT  #EM$SSY         ;Can't do this to SY
                                ;
                                ;  See if this device is mounted by anyone
                                ;
004330  004767  000000G         11$:    CALL    CHKMNT          ;See if device is mounted by anyone
004334  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
                                ;
004336  004767  000000G                 CALL    CHKMTX          ;Is device mounted by any other users?
004342  103004                          BCC     2$              ;Br if not
004344                          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.
                                ;
004354  012700  000000G        2$:     MOV     #MNTARG,R0      ;Remount the device
004360  104375                          EMT     375
                                ;
                                ;  See if this device is allocated to any other user
                                ;
004362  120467  000000G        8$:     CMPB    R4,LDDEVX       ;Is this a logical disk?
004366  001404                          BEQ     12$             ;Br if yes -- Don't check for allocation
004370  016700  000000G                 MOV     ALCDEV,R0       ;Get RAD50 device name
004374  004767  000000G                 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.
                                ;
004400  016705  173374         12$:     MOV     SQZDEV,R5       ;Get back device name
004404  004767  000000G                 CALL    CHKDEV          ;Convert device name to dev # & unit #
004410  020467  000000G                 CMP     R4,LDDEVX       ;Is this a logical disk?
004414  001406                          BEQ     4$              ;Br if yes
004416  005002                          CLR     R2              ;Say base block # = 0
004420  012703  177777                 MOV     #177777,R3      ;Say top block = 177777
004424  000300                          SWAB    R0              ;Put unit # in high byte
004426  050004                          BIS     R0,R4           ;Get unit # and device # together
004430  000410                          BR      5$
004432  006300                 4$:      ASL     R0              ;Convert unit # to word table index
004434  016004  000000G                 MOV     LDPDEV(R0),R4   ;Get physical dev # and unit #
004440  016002  000000G                 MOV     LDBASE(R0),R2   ;Get base block of logical disk
004444  010203                          MOV     R2,R3
004446  066003  000000G                 ADD     LDSIZE(R0),R3   ;Get top block number of logical disk
                                ;
                                ;  Search for any mounted logical disks that are on the disk being squeezed
                                ;
004452  016705  000000G        5$:     MOV     CSHDEV,R5       ;Point to start of mount table
004456  010500                 7$:     MOV     R5,R0           ;Get address of mount entry
004460  004767  000000G                 CALL    CDGET           ;Copy mount entry into CDBUF
004464  020467  000001C                 CMP     R4,CDBUF+CD$DVU ;Is this device on same physical device?
004470  001014                          BNE     6$              ;Br if not
004472  016700  000001C                 MOV     CDBUF+CD$BAS,R0 ;Is this a logical disk?
004476  001411                          BEQ     6$              ;Br if not
004500  020002                          CMP     R0,R2           ;See if logical disk is within disk being sqze
004502  101407                          BLOS    6$              ;Br if not
004504  020003                          CMP     R0,R3           ;Compare upper range
004506  103005                          BHIS    6$
                                ;
                                ;  We found a logical disk within the disk being squeezed.
                                ;  See if anyone else has that logical disk mounted.
                                ;
004510  004767  000000G                 CALL    CHKMTX          ;Anyone else have this logical disk mounted?
004514  103713                          BCS     3$              ;Error if yes
                                ;
                                ;  Dismount this logical disk and any files associated with it
                                ;
004516  004767  000000G                 CALL    DMTSUB          ;Do the dismount
                                ;
                                ;  Check for more nested logical disks
                                ;
```

```
    115 004522  062705  000000G         6$:     ADD     #CD$$SZ,R5      ;Point to next mount table entry
    116 004526  020567  000000G                 CMP     R5,CSHDVN       ;Reached end of table?
    117 004532  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 004534  016705  173240                  MOV     SQZDEV,R5       ;Get name of device being squeezed
    123 004540  004767  000000G                 CALL    LOGCHK          ;See if log file is on that device
    124                                 ;
    125                                 ;  Call CCL to process the command
    126                                 ;
    127 004544  000167  000000G         7$:     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
                                ;
004550  010246                  CKSYDV: MOV     R2,-(SP)
004552  010346                          MOV     R3,-(SP)
004554  010446                          MOV     R4,-(SP)
004556  010546                          MOV     R5,-(SP)
                                ;
                                ;  Convert device name to device index and unit number
                                ;
004560  004767  0000000G                CALL    CHKDEV          ;Convert name to device index & unit #
004564  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.
                                ;
004566  012705  004646'                 MOV     #SYFLVC,R5      ;Point to vector of savestatus addresses
004572  012503                  1$:     MOV     (R5)+,R3        ;Point to next savestatus block
004574  001416                          BEQ     8$              ;Br if reached end of list
004576  016302  0000000G                MOV     C.CSW(R3),R2    ;Get 1st word of channel
004602  042702  177701                  BIC     #^C<76>,R2      ;Extract device index number
004606  020204                          CMP     R2,R4           ;Does it match device being squeezed?
004610  001370                          BNE     1$              ;Br if not
004612  116302  0000000G                MOVB    C.DEVQ(R3),R2   ;Get unit number
004616  042702  177770                  BIC     #^C<7>,R2       ;Extract unit number
004622  020200                          CMP     R2,R0           ;Same as unit of device being squeezed?
004624  001362                          BNE     1$              ;Br if not
                                ;
                                ;  This device contains a system file
                                ;
004626  000261                          SEC                     ;Signal error on return
004630  000401                          BR      9$
                                ;
                                ;  This device does not contain a system file
                                ;
004632  000241                  8$:     CLC                     ;Signal no error
                                ;
                                ;  Finished
                                ;
004634  012605                  9$:     MOV     (SP)+,R5
004636  012604                          MOV     (SP)+,R4
004640  012603                          MOV     (SP)+,R3
004642  012602                          MOV     (SP)+,R2
004644  000207                          RETURN
                                ;
                                ;  List of addresses of savestatus blocks for system files
                                ;
004646  0000000G                SYFLVC: .WORD   SWPCHN           ;Swap file
004650  0000000G                        .WORD   SEGCHN           ;PLAS swap file
004652  0000000G                        .WORD   SPLCHN           ;Spool file
```

```
     58 004654  000000G                          .WORD   KMNCHN          ;TSKMON.SAV
     59 004656  000000G                          .WORD   CCLSAV          ;CCL.SAV
     60 004660  000000G                          .WORD   INDSAV          ;IND.SAV
     61 004662  000000G                          .WORD   INDTSV          ;TSXIND.TSX
     62 004664  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
                                ;
004666  005067  000000G        CMDMON: CLR     MNFLGS          ;INITIALIZE EMT ARG BLOCK
004672  005067  000000G                CLR     MNBPC
004676  004767  000000G                CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACES
004702  004767  000000G                CALL    DELSPC          ;DELETE SPACES FROM COMMAND LINE
004706  004767  000000G                CALL    ACROCT          ;ACCRUE BASE ADDRESS
004712  010167  000000G                MOV     R1,MNBASE       ;SET IN EMT ARG BLOCK
004716  122327  000054                CMPB    (R3)+,#',       ;COMMA SHOULD BE DELIMITER
004722  001075                        BNE     3$
004724  004767  000000G                CALL    ACROCT          ;ACCRUE TOP ADDRESS
004730  010167  000000G                MOV     R1,MNTOP
004734  112300                        MOVB    (R3)+,R0        ;GET DELIMITER
004736  001430                        BEQ     1$              ;BR IF FINISHED WITH COMMAND
004740  120027  000054                CMPB    R0,#',          ;DID HE SPECIFY NBPC?
004744  001006                        BNE     2$              ;BR IF NOT
004746  004767  000000G                CALL    ACRDEC          ;ACCRUE NUMBER OF BYTES PER CELL
004752  010167  000000G                MOV     R1,MNBPC
004756  112300                4$:     MOVB    (R3)+,R0        ;GET DELIMITER
004760  001417                        BEQ     1$              ;BR IF END OF COMMAND HIT
004762  120027  000057        2$:     CMPB    R0,#'/          ;GOT A SWITCH?
004766  001053                        BNE     3$              ;BR IF NOT
004770  012704  000234'              MOV     #MONHD,R4       ;POINT TO TABLE OF SWITCH OPTIONS
004774  004767  000000G                CALL    SEARCH          ;SEARCH FOR SWITCH IN TABLE
005000  103004                        BCC     5$              ;BR IF FOUND IT
005002                                FABORT  #INVOPT         ;INVALID SWITCH
005012  051467  000000G        5$:     BIS     (R4),MNFLGS     ;SET BITS FOR SWITCH
005016  000757                        BR      4$              ;GO CHECK FOR MORE SWITCHES
                                ;
                                ;  Finished scanning command.
                                ;  Initiate performance monitoring.
                                ;
005020  012700  000000G        1$:     MOV     #MONAR1,R0      ;POINT TO EMT ARG BLOCK
005024  104375                        EMT     375             ;TRY TO INITIATE MONITORING
005026  103026                        BCC     6$              ;BR IF OK
005030  105737  000000G                TSTB    @#ERRLOC        ;CHECK ERROR CODE
005034  001404                        BEQ     7$              ;BR IF SOMEONE ELSE DOING MONITORING NOW
005036                                FABORT  #NOPMGN         ;PM NOT GENNED IN
005046  010046                7$:     MOV     R0,-(SP)        ;SAVE # OF PM USER
005050                                FERR    #PMBUSY         ;PM FACILITY IN USE BY ANOTHER USER
005064  012605                        MOV     (SP)+,R5        ;GET # OF PM USER
005066  004767  000000G                CALL    PRTDEC          ;PRINT DECIMAL VALUE
005072                                .PRINT  #CRLF           ;TERMINATE THE ERROR LINE
005100  000167  000000G                JMP     RDCMD
005104  012700  000000G        6$:     MOV     #MONAR2,R0      ;START MONITORING
005110  104375                        EMT     375
005112  000167  000000G                JMP     RDCMD
                                ;
                                ;  Invalid command syntax.
                                ;
005116                          3$:     FABORT  #CSIMS1
```

```
                                        .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
                                ;
005126  004767  000000G        CMDSPO: CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACE
005132  004767  000000G                CALL    CVTUC           ;Convert lower case chars to upper case
005136  105713                        TSTB    (R3)            ;ANY DEVICE NAMED?
005140  001002                        BNE     1$              ;BRANCH IF YES
005142  000167  000000G               JMP     BADCMD          ;MUST HAVE DEVICE NAME
                                ;
                                ;  ACCRUE DEVICE NAME IN RAD50 FORM.
                                ;
005146  004767  000000G        1$:     CALL    GTRD50          ;ACCRUE THE NAME
                                ;
                                ;  If device name is a logical name, translate to physical
                                ;
005152  016700  000000G                MOV     R50BUF,R0       ;Get device name
005156  004767  000000G                CALL    ASNSRC          ;See if this name is assigned
005162  103403                        BCS     3$              ;Br if name is not assigned
005164  016267  000000G 000000G        MOV     AT$DEV(R2),R50BUF;Set physical device name
                                ;
                                ;  NOW TRY TO FIND SDCB WHICH MATCHES NAME.
                                ;
005172  010346                 3$:     MOV     R3,-(SP)        ;SAVE POINTER
005174  012703  000000G                MOV     #R50BUF,R3      ;POINT TO DEVICE NAME
005200  004767  000000G                CALL    FORCE0          ;MAKE UNIT 0 IF NEEDED
005204  012603                        MOV     (SP)+,R3        ;RESTORE POINTER
005206  012701  000000G                MOV     #SDCB,R1        ;POINT TO 1ST SDCB
005212  020127  000000G        4$:     CMP     R1,#SDCBND      ;CHECKED ALL?
005216  103013                        BHIS    2$              ;BRANCH IF YES
005220  026127  000000G 000000G        CMP     SDNAME(R1),#DMYDEV;Is this dummy device entry?
005226  001404                        BEQ     5$              ;Br if yes
005230  026761  000000G 000000G        CMP     R50BUF,SDNAME(R1) ;DO NAMES MATCH?
005236  001407                        BEQ     CSGTOP          ;BRANCH IF FOUND SDCB
005240  062701  000000G        5$:     ADD     #SDCBSZ,R1      ;GO CHECK NEXT SDCB
005244  000762                        BR      4$
                                ;
                                ;  CAN'T FIND SDCB FOR DEVICE.
                                ;
005246                          2$:     FABORT  #INVDEV
                                ;
                                ;  FOUND SDCB (R1).
                                ;  IDENTIFY OPTION.
                                ;
005256  112300                  CSGTOP: MOVB    (R3)+,R0        ;SKIP OVER ANY DELIMITERS
005260  001424                          BEQ     3$              ;Br at end of command string
005262  004767  000000G                CALL    CHKDLM
005266  103773                        BCS     CSGTOP
005270  005303                        DEC     R3
005272  012704  000260'               MOV     #SPLHD,R4       ;POINT TO TABLE OF SPOOL OPTIONS
005276  004767  000000G                CALL    SEARCH          ;LOOK UP OPTION
005302  103401                        BCS     1$              ;BR IF ERROR
                                ;
                                ;  FOUND OPTION.   BRANCH OFF TO OPTION PROCESSING ROUTINE.
```

```
                                ;  SDCB ADDRESS IS IN R1.
                                ;
005304  000134                        JMP     @(R4)+
                                ;
                                ;  ERROR IN SEARCH
                                ;
005306  005704                  1$:   TST     R4              ;ILLEGAL OR AMBIGUOUS
005310  001404                        BEQ     2$              ;BR IF ILLEGAL
005312                                FABORT  #AMBOPT         ;AMBIGUOUS OPTION
005322                          2$:   FABORT  #INVOPT         ;ILLEGAL OPTION
                                ;
                                ;  Reached end of command string (usually when handling multiple options)
                                ;
005332  000167  000000G         3$:   JMP     RDCMD
```

```
                                ;.....................................................
                                ;
                                ;  PROCESS THE 'DEL' (DELETE) OPTION.
                                ;
005336                          SPLDEL:
                                ;
                                ;  See if a file ID was specified
                                ;
005336  105713                          TSTB    (R3)            ;Was an id specified?
005340  001425                          BEQ     SPDLCR          ;Br if not
005342  121327  000054                  CMPB    (R3),#',        ;Comma used as delimiter?
005346  001403                          BEQ     1$              ;Br if yes
005350  121327  000075                  CMPB    (R3),#'=        ;Equal sign used as delimiter?
005354  001001                          BNE     2$              ;Br if not -- space must have been delimiter
005356  005203                  1$:     INC     R3              ;Skip over comma or equal sign
                                ;
                                ;  A file id was specified, accrue the id value
                                ;
005360  004767  000000G         2$:     CALL    ACRDEC          ;Accrue the value
005364  010167  172462                  MOV     R1,DELEMT+4     ;Set file ID in EMT arg block
005370  012700  000046'                  MOV     #DELEMT,R0      ;Point to EMT arg block
005374  104375                          EMT     375             ;Try to delete the file
005376  103004                          BCC     9$              ;Br if successful
005400                                  FABORT  #EM$NID         ;Cannot find file with that id
005410  000167  000000G         9$:     JMP     RDCMD
                                ;
                                ;  Abort the file currently being printed by the spooled device
                                ;
005414  005761  000000G         SPDLCR: TST     SDSFCB(R1)      ;IS DEVICE BUSY?
005420  001405                          BEQ     SPNBSY          ;BRANCH IF NOT
005422  152761  000000G 000000G         BISB    #SD$DEL,SDFLAG(R1) ;SET DELETE FLAG
005430  000167  000000G         CSABT:  JMP     RDCMD
005434                          SPNBSY: .PRINT  #DEVIDL
005442  000772                          BR      CSABT
```

```
                                ;*******************************************************************************
                                ;  PROCESS THE SPOOL 'ALIGN' OPTION.
                                ;  (SPOOL XX,ALIGN,<FILE NAME>)
                                ;
005444  004767  000000G        SPLALN: CALL    CKPRIV          ;Must have operator privilege
005450  122327  000054                BCMPB   (R3)+,#',       ;WAS FILE NAME GIVEN?
005454  001404                        BEQ     1$              ;BR IF YES
005456                          9$:     FABORT  #COAL
                                ;  PARSE ALIGNMENT FILE NAME
005466  010605                  1$:     MOV     SP,R5           ;SAVE SP
005470                                  .CSIGEN #ALDEX,#ALDEX,R3;OPEN ALIGNEMENT FILE - CHAN 3
005504  010506                        MOV     R5,SP           ;RESET SP
005506  103763                        BCS     9$              ;BRANCH IF .CSIGEN ERROR
                                ;  ALIGNEMENT FILE IS NOW OPEN ON CHANNEL 3
                                ;  OPEN SPOOLED DEVICE ON CHANNEL 1
                                ;
005510  016167  000000G 000000G       MOV     SDNAME(R1),ALDBLK;SET UP DEVICE NAME
005516                                  .ENTER  #XAREA,#1,#ALDBLK,#0;OPEN SPOOLED DEVICE
005542  103010                        BCC     2$              ;BR IF OPEN OK
005544                                  .PRINT  #COAD           ;CAN'T OPEN SPOOLED DEVICE
005552                                  .CLOSE  #3              ;CLOSE ALIGNMENT FILE
005560  000167  000000G        10$:    JMP     RDCMD
                                ;
                                ;  WRITE 1ST RECORD TO SPOOLED FILE AND SPECIFY A
                                ;  FORM NAME OF "******" WHICH MEANS ALIGNMENT FILE.
                                ;
005564                          2$:     .WRITW  #XAREA,#1,#ALFN,#4,#0
                                ;
                                ;  NOW COPY ALIGNMENT FILE TO SPOOLED FILE
                                ;
005622  005002                        CLR     R2              ;INIT BLOCK #
005624                          4$:     .READW  #XAREA,#3,#BLKO,#256.,R2
005662  103421                        BCS     3$              ;BR IF END OF FILE
005664                                  .WRITW  #XAREA,#1,#BLKO,#256.,R2
005722  005202                        INC     R2
005724  000737                        BR      4$
                                ;  HIT END OF ALIGNMENT FILE.   CLOSE BOTH CHANNELS.
005726                          3$:     .CLOSE  #1
005734                                  .CLOSE  #3
005742  000706                        BR      10$
                                ;
                                ;  PROCESS THE 'LOCK' OPTION.
                                ;
005744  004767  000000G        SPLLK:  CALL    CKPRIV          ;Must have operator privilege
005750  052761  000000G 000000G       BIS     #SD$FLK,SDFLAG(R1);SAY FORM IS LOCKED
005756  000405                        BR      SPLMT1          ;NOW DO MOUNT
                                ;
                                ;  PROCESS THE 'FORM' OPTION
                                ;
005760  004767  000000G        SPLFRM: CALL    CKPRIV          ;Must have operator privilege
005764  042761  000000G 000000G       BIC     #SD$FLK,SDFLAG(R1);SAY FORM IS NOT LOCKED
005772  010102                  SPLMT1: MOV     R1,R2           ;POINT TO FORM NAME CELL
005774  062702  000000G                ADD     #SDFORM,R2
006000  010204                        MOV     R2,R4           ;POINT TO END OF CELL
006002  062704  000006                ADD     #6,R4
006006  005203                        INC     R3              ;SKIP OVER COMMA
006010  105713                  3$:   TSTB    (R3)            ;HIT END OF NAME?
```

```
      58 006012  001404                          BEQ     1$              ;BR IF YES
      59 006014  020204                          CMP     R2,R4           ;COPIED 6 CHARS?
      60 006016  103007                          BHIS    2$              ;BR IF YES
      61 006020  112322                          MOVB    (R3)+,(R2)+     ;COPY IN FORM NAME
      62 006022  000772                          BR      3$
      63                                ;  PAD END OF SHORT NAME WITH BLANKS
      64 006024  020204                 1$:      CMP     R2,R4           ;FILLED TO 6 CHARS?
      65 006026  103003                          BHIS    2$              ;BR IF YES
      66 006030  112722  000040                  MOVB    #' ,(R2)+       ;PUT IN TRAILING SPACES
      67 006034  000773                          BR      1$
      68                                ;
      69                                ;  NOW TRY TO START SPOOLER
      70                                ;
      71 006036  042761  000000G 000000G 2$:      BIC     #SD$WFM,SDFLAG(R1);SAY FORM MOUNT IS DONE
      72 006044  010100                          MOV     R1,R0           ;POINT TO SDCB
      73 006046  004767  000630                  CALL    SPOLGO          ;TRY TO START SPOOLER
      74 006052  000167  000000G                 JMP     RDCMD
      75                                ;
      76                                ;  PROCESS THE 'SKIP' OPTION
      77                                ;
      78 006056  004767  000000G        SPLSKP: CALL    CKPRIV          ;Must have operator privilege
      79 006062  005761  000000G                 TST     SDSFCB(R1)      ;IS DEVICE BUSY?
      80 006066  001415                          BEQ     SPNBB           ;BR IF NOT
      81                                ; ACCRUE NUMBER
      82 006070  010104                          MOV     R1,R4           ;SAVE SDCB ADDRESS
      83 006072  005203                          INC     R3              ;SKIP OVER COMMA
      84 006074  004767  000000G                 CALL    ACRDEC          ;ACCRUE DEC NUMBER
      85 006100  010164  000000G                 MOV     R1,SDSKIP(R4)   ;SET SKIP COUNT IN SDCB
      86 006104  000167  000000G                 JMP     RDCMD
      87                                ;
      88                                ;  PROCESS THE 'BACK' OPTION
      89                                ;
      90 006110  004767  000000G        SPLBAK: CALL    CKPRIV          ;Must have operator privilege
      91 006114  005761  000000G                 TST     SDSFCB(R1)      ;IS THE DEVICE BUSY
      92 006120  001002                          BNE     SPLBK1          ;BR IF YES
      93 006122  000167  177306         SPNBB:  JMP     SPNBSY          ;GO GIVE ERROR MESSAGE
      94 006126  005761  000000G        SPLBK1: TST     SDBU(R1)        ;ANY BLOCKS REMEMBERED YET?
      95 006132  001403                          BEQ     1$              ;BR IF NOTHING TO BACKUP TO
      96 006134  052761  000000G 000000G         BIS     #SD$BAK,SDFLAG(R1);REQUEST BACK UP
      97 006142  000167  000000G        1$:     JMP     RDCMD
```

```
                                        ;
                                        ;  PROCESS THE SPOOL 'STATUS' COMMAND
                                        ;
006146  005761  000000G       SPLSTA:   TST     SDSFCB(R1)          ;IS SPOOLER BUSY?
006152  001404                          BEQ     1$                  ;BR IF NOT
006154                                  .PRINT  #SPACTV
006162  000413                          BR      4$
006164  032761  000000G 000000G 1$:     BIT     #SD$WFM,SDFLAG(R1);IS IT WAITING FOR A FORM MOUNT?
006172  001404                          BEQ     2$
006174                                  .PRINT  #SPWFM
006202  000403                          BR      4$
006204                          2$:     .PRINT  #DEVIDL
                                        ;  SEE IF IN SINGLE FILE MODE
006212  032761  000000G 000000G 4$:     BIT     #SD$SNG,SDFLAG(R1);IN SINGLE-FILE MODE?
006220  001403                          BEQ     20$                 ;BR IF NOT
006222                                  .PRINT  #SPSNG
                                        ;  SEE IF SPOOL FILE IS FULL
006230  005767  000000G       20$:     TST     NFRESB              ;IS SPOOL FILE FULL?
006234  001003                          BNE     3$                  ;BR IF NOT
006236                                  .PRINT  #SPFUL
                                        ;  LIST CURRENT FORM NAME
006244                          3$:     .PRINT  #SPCF
006252  010102                          MOV     R1,R2               ;POINT TO FORM NAME CELL
006254  062702  000000G                 ADD     #SDFORM,R2
006260  012703  000006                  MOV     #6,R3               ;PRINT 6 CHARS
006264  112200                  5$:     MOVB    (R2)+,R0
006266                                  .TTYOUT
006272  005303                          DEC     R3
006274  001373                          BNE     5$
                                ;  SEE IF CURRENT FORM IS LOCKED
006276  032761  000000G 000000G         BIT     #SD$FLK,SDFLAG(R1);IS FORM LOCKED?
006304  001403                          BEQ     21$                 ;BR IF NOT
006306                                  .PRINT  #SPFLK
                                        ;  SEE IF IN FLAG PAGE MODE
006314  112700  000054        21$:     MOVB    #',,R0              ;PRINT COMMA
006320                                  .TTYOUT
006324  032761  000000G 000000G         BIT     #SD$FLG,SDFLAG(R1);IN FLAG PAGE MODE?
006332  001004                          BNE     6$                  ;BR IF YES
006334                                  .PRINT  #NSPFLG             ;"NOFLAGPAGE"
006342  000416                          BR      7$
006344                          6$:     .PRINT  #SPFLG              ;"FLAGPAGE,"
006352  032761  000000G 000000G         BIT     #SD$WID,SDFLAG(R1);IS FLAG PAGE WIDE?
006360  001004                          BNE     61$                 ;BR IF YES
006362                                  .PRINT  #SPNARO             ;"NARROW"
006370  000403                          BR      7$
006372                          61$:    .PRINT  #SPWIDE             ;"WIDE"
                                ;  List info about files pending for this device.
006400  005761  000000G         7$:     TST     SDFHD(R1)           ;ARE THERE ANY FILES PENDING FOR THIS DEVICE?
006404  001004                          BNE     11$                 ;BR IF YES
006406                                  .PRINT  #NOFIL              ;NO FILES...
006414  000410                          BR      8$
006416                          11$:    .PRINT  #QHDMS1             ;PRINT HEADING MESSAGE
006424                                  .PRINT  #QHDMS2             ;UNDERLINE THE HEADING
006432  004767  000000G                 CALL    LSTSPL              ;LIST SPOOL FILES
006436  000167  000000G         8$:     JMP     RDCMD
                                        ;
                                        ;  PROCESS THE 'SINGLE' OPTION
```

```
     58                                     ;
     59 006442  004767  0000000G            SPLSNG: CALL    CKPRIV          ;Must have operator privilege
     60 006446  052761  0000000G 0000000G           BIS     #SD$SNG,SDFLAG(R1);SET SINGLE-FILE MODE
     61 006454  000167  0000000G                     JMP     RDCMD
     62                                     ;
     63                                     ;  PROCESS THE 'MULTIPLE' OPTION
     64                                     ;
     65 006460  004767  0000000G            SPLMUL: CALL    CKPRIV          ;Must have operator privilege
     66 006464  042761  0000000G 0000000G           BIC     #SD$SNG,SDFLAG(R1);RESET SINGLE-FILE MODE
     67 006472  000167  0000000G                     JMP     RDCMD
     68                                     ;
     69                                     ;  PROCESS THE SPOOL 'HOLD' COMMAND.
     70                                     ;  HOLD MEANS DON'T START OUTPUT UNTIL CHANNEL IS CLOSED.
     71                                     ;
     72 006476  004767  0000000G            SPLHLD: CALL    CKPRIV          ;Must have operator privilege
     73 006502  010167  171352                       MOV     R1,SPHLEM+4     ;Set SDCB address in EMT arg block
     74 006506  012700  000054'                      MOV     #SPHLEM,R0      ;Point to EMT arg block
     75 006512  104375                               EMT     375             ;Set hold mode
     76 006514  000167  0000000G                     JMP     RDCMD
     77                                     ;
     78                                     ;  PROCESS THE SPOOL 'NOHOLD' COMMAND WHICH MEANS START OUTPUT
     79                                     ;  AS SOON AS A FILE IS CREATED.
     80                                     ;
     81 006520  004767  0000000G            SPLNHL: CALL    CKPRIV          ;Must have operator privilege
     82 006524  010167  171336                       MOV     R1,SPNHEM+4     ;Set SDCB address in EMT arg block
     83 006530  012700  000062'                      MOV     #SPNHEM,R0      ;Point to EMT arg block
     84 006534  104375                               EMT     375             ;Set NOHOLD mode
     85 006536  000167  0000000G                     JMP     RDCMD
     86                                     ;
     87                                     ;  PROCESS THE 'FLAGPAGE' OPTION
     88                                     ;
     89 006542  004767  0000000G            SPLFLG: CALL    CKPRIV          ;Must have operator privilege
     90 006546  052761  0000000G 0000000G           BIS     #SD$FLG,SDFLAG(R1);SET FLAG MODE
     91 006554  000167  176476                       JMP     CSGTOP          ;Go check for other options
     92                                     ;
     93                                     ;  PROCESS THE 'NOFLAGPAGE' OPTION
     94                                     ;
     95 006560  004767  0000000G            SPLNFL: CALL    CKPRIV          ;Must have operator privilege
     96 006564  042761  0000000G 0000000G           BIC     #SD$FLG,SDFLAG(R1);RESET FLAG MODE
     97 006572  000167  176460                       JMP     CSGTOP          ;Go check for other options
     98                                     ;
     99                                     ;  PROCESS THE 'WIDE' OPTION
    100                                     ;
    101 006576  004767  0000000G            SPLWID: CALL    CKPRIV          ;Must have operator privilege
    102 006602  052761  0000000G 0000000G           BIS     #SD$WID,SDFLAG(R1);SET FLAG MODE
    103 006610  000167  176442                       JMP     CSGTOP          ;Go check for other options
    104                                     ;
    105                                     ;  PROCESS THE 'NARROW' OPTION
    106                                     ;
    107 006614  004767  0000000G            SPLNRW: CALL    CKPRIV          ;Must have operator privilege
    108 006620  042761  0000000G 0000000G           BIC     #SD$WID,SDFLAG(R1);RESET FLAG MODE
    109 006626  000167  176424                       JMP     CSGTOP          ;Go check for other options
    110                                     ;
    111                                     ;-----------------------------------------------------
    112                                     ;  PROCESS THE 'FORM' COMMAND.
    113                                     ;
    114 006632  004767  0000000G            CMDFRM: CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACE
```

```
     115 006636  004767  000000G                 CALL    CVTUC           ;CONVERT ALL CHARS TO UPPER CASE
     116 006642  012704  000000G                 MOV     #UFORM,R4       ;POINT TO CELL WHICH HOLDS FORM NAME
     117 006646  020302                  2$:     CMP     R3,R2           ;MOVED ALL OF NAME?
     118 006650  103004                          BHIS    1$              ;BR IF YES
     119 006652  112324                          MOVB    (R3)+,(R4)+     ;MOVE FORM NAME TO UFORM
     120 006654  020427  000006G                 CMP     R4,#<UFORM+6>   ;MOVED 6 CHARS?
     121 006660  103772                          BLO     2$              ;BR IF NOT
     122                                 ;  SEE IF WE NEED TO PAD NAME WITH BLANKS
     123 006662  020427  000006G         1$:     CMP     R4,#<UFORM+6>   ;FILLED TO 6 CHARS?
     124 006666  103003                          BHIS    3$              ;BR IF FINISHED
     125 006670  112724  000040                  MOVB    #' ,(R4)+       ;PUT IN TRAILING BLANKS
     126 006674  000772                          BR      1$
     127 006676  000167  000000G         3$:     JMP     RDCMD
     128
     129                                 ;-------------------------------------------------------------------------
     130                                 ;  SPOLGO is called to start a spooler.
     131                                 ;
     132                                 ;  Inputs:
     133                                 ;   R0 = SDCB address of spooler to be started.
     134                                 ;
     135 006702  010046                  SPOLGO: MOV     R0,-(SP)
     136 006704  010067  000044G                 MOV     R0,SPGEMT+4     ;SET SDCB ADDRESS IN EMT ARG BLOCK
     137 006710  012700  000000G                 MOV     #SPGEMT,R0      ;POINT TO ARG BLOCK
     138 006714  104375                          EMT     375             ;START THE SPOOLER
     139 006716  012600                          MOV     (SP)+,R0
     140 006720  000207                          RETURN
```

```
                                        .SBTTL  SEND command
                                ;-------------------------------------------------------------------------
                                ;  Process the YELL command which overrides TT gag but requires
                                ;  operator privilege.
                                ;
006722  004767  000000G        CMDYEL: CALL    CKPRIV          ;Require OPER privilege
006726  112767  000003  000000G       MOVB    #3,YELEMT       ;Set the YELL and error flags
006734  000411                       BR      SNDCOM          ;Enter SEND command code

                                ;-----------------------------------------------------
                                ;  Process the OPERATOR command which functions as a send
                                ;  command to the operators console.
                                ;
006736  116701  000000G        OPRCMD: MOVB    CTRLTT,R1       ;GET LINE INDEX # OF OPR TERMINAL
006742  001014                       BNE     SNDGN           ;BR IF GOT ONE
006744                               FABORT  #NOOPTT         ;NO OPERATOR'S CONSOLE

                                ;-----------------------------------------------------------
                                ;  PROCESS THE 'SEND' COMMAND.
                                ;
006754  105067  000000G        CMDSND: CLRB    YELEMT          ;This is not the YELL command
006760  111300                 SNDCOM: MOVB    (R3),R0         ;GET 1ST CHAR OF COMMAND
006762  001420                       BEQ     CSEXIT          ;BRANCH IF NULL COMMAND
                                ;
                                ;  ACCRUE LINE #.
                                ;
006764  004767  000544               CALL    SNDLIN          ;Accrue line # to send to
006770  005701                       TST     R1              ;SEND TO SPECIFIC USER?
006772  003422                       BLE     CSSCAN          ;BRANCH IF ALL OR OC
006774  020127  000000G        SNDGN:  CMP     R1,#LSTSL       ;IS THIS A VALID LINE #?
007000  101003                       BHI     LNUMER          ;BR IF NOT
007002  016101  000000G              MOV     LNPRIM(R1),R1   ;GET PRIMARY LINE #
007006  001010                       BNE     CSCD            ;BRANCH IF VALID
007010                         LNUMER: FWARN   #BDLIN          ;Invalid line number
007024  000167  000000G        CSEXIT: JMP     RDCMD
007030  032761  000000G 000000G CSCD:   BIT     #$DETCH,LSW(R1) ;TRYING TO SEND TO DETACHED JOB?
007036  001364                       BNE     LNUMER          ;BR IF YES
                                ;
                                ;  AT THIS POINT THE INDEX NUMBER OF THE LINE WE WISH TO
                                ;  SEND TO IS IN R1.
                                ;  PUT OUR LINE NUMBER AND USER NAME AT START OF MESSAGE BUFFER
                                ;
007040  032767  000000G 000000G CSSCAN: BIT     #PO$SND,PRIVCO  ;Are we privileged to send messages?
007046  001004                       BNE     10$             ;Br if yes
007050                               FABORT  #EM$SND
007060  012704  000000G        10$:    MOV     #MSGBUF,R4      ;POINT TO MESSAGE BUFFER
007064  112724  000015               MOVB    #CR,(R4)+       ;PUT IN LEADING CR-LF-BELL
007070  112724  000012               MOVB    #LF,(R4)+
007074  112724  000007               MOVB    #BELL,(R4)+
007100  116705  000000G              MOVB    CORUSR,R5       ;GET OUR LINE NUMBER
007104  016505  000000G              MOV     LNPRIM(R5),R5   ;GET OUR PRIMARY LINE NUMBER
007110  010502                       MOV     R5,R2
007112  062702  000000G              ADD     #NUMTB1,R2      ;POINT TO TABLE WITH LINE NUMBER CHARACTERS
007116  112224                       MOVB    (R2)+,(R4)+     ;PUT IN LINE NUMBER
007120  112224                       MOVB    (R2)+,(R4)+
007122  070527  000006               MUL     #6,R5           ;EACH JOB HAS A 12 CHAR USER NAME
007126  062705  000000G              ADD     #LUNAME,R5      ;POINT TO USER NAME ENTRY FOR THIS LINE
```

```
     58 007132  121527  000040                    CMPB    (R5),#'         ;IS USER NAME BLANK?
     59 007136  001416                            BEQ     4$              ;BR IF YES
     60 007140  112724  000040                    MOVB    #' ,(R4)+       ;PUT IN A SPACE
     61 007144  112724  000050                    MOVB    #'(,(R4)+       ;AND OPEN PAREN AT START OF NAME
     62 007150  012702  000014                    MOV     #12.,R2         ;GET LENGTH OF NAME
     63 007154  112524                  6$:       MOVB    (R5)+,(R4)+     ;GET NEXT CHAR OF USER NAME
     64 007156  077202                            SOB     R2,6$           ;MOVE REST OF NAME
     65 007160  124427  000040          7$:       CMPB    -(R4),#BLANK    ;TRIM OFF TRAILING SPACES
     66 007164  001775                            BEQ     7$
     67 007166  005204                            INC     R4              ;POINT BEYOND LAST NON-BLANK CHARACTER
     68 007170  112724  000051          5$:       MOVB    #'),(R4)+       ;PUT IN CLOSE PAREN AT END OF NAME
     69 007174  112724  000040          4$:       MOVB    #' ,(R4)+       ;PUT " -- " AT END OF NAME
     70 007200  112724  000055                    MOVB    #'-,(R4)+
     71 007204  112724  000055                    MOVB    #'-,(R4)+
     72 007210  112724  000040                    MOVB    #' ,(R4)+
     73                                  ;
     74                                  ;  Now move message string to message buffer
     75                                  ;
     76 007214  122327  000040          3$:       CMPB    (R3)+,#'        ;SKIP OVER ANY BLANKS
     77 007220  001775                            BEQ     3$
     78 007222  005303                            DEC     R3              ;POINT TO 1ST CHAR OF MESSAGE
     79 007224  112324                  1$:       MOVB    (R3)+,(R4)+     ;MOVE MESSAGE TO OUR BUFFER
     80 007226  001403                            BEQ     2$              ;BRANCH WHEN END HIT
     81 007230  020427  177776G                   CMP     R4,#<MSGEND-2>  ;DON'T OVERFLOW OUR BUFFER
     82 007234  103773                            BLO     1$
     83 007236  005304                  2$:       DEC     R4              ;PUT CR-LF AT END OF MESSAGE
     84 007240  112724  000015                    MOVB    #CR,(R4)+
     85 007244  112724  000012                    MOVB    #LF,(R4)+
     86 007250  105014                            CLRB    (R4)            ;PUT IN NULL TO SIGNAL END
     87                                  ;
     88                                  ;  See who we should send the message to.
     89                                  ;
     90 007252  005701                            TST     R1              ;WHO IS IT GOING TO?
     91 007254  001655                            BEQ     LNUMER          ;SEND TO CONSOLE TERMINAL
     92 007256  100450                            BMI     SNDALL          ;SEND TO ALL LINES
     93                                  ;
     94                                  ;  Send message to one terminal whose line index number
     95                                  ;  is in R1.
     96                                  ;
     97 007260  032761  000000G 000000G SNDONE:   BIT     #$DILUP,LSW(R1) ;IS THAT LINE LOGGED ON?
     98 007266  001005                            BNE     1$              ;BRANCH IF YES
     99 007270                                    .PRINT  #NOTON
    100 007276  000167  000000G                   JMP     RDCMD
    101 007302  032761  000000G 000000G 1$:      BIT     #$TTGAG,LSW/(R1);IS LINE GAGGED?
    102 007310  001414                            BEQ     2$              ;BR IF NOT
    103 007312  032761  000000G 000000G          BIT     #$INKMN,LSW4(R1);IS JOB IN KMON?
    104 007320  001010                            BNE     2$              ;BR IF YES
    105 007322  105767  000000G                   TSTB    YELEMT          ;Are we yelling?
    106 007326  001005                            BNE     2$              ;Br if yes -- Override gag
    107 007330                                    .PRINT  #GAGMSG         ;SAY LINE IS GAGGED
    108 007336  000167  000000G                   JMP     RDCMD
    109 007342  012700  000000G          2$:      MOV     #YELEMT,R0      ;POINT TO EMT ARG BLOCK
    110 007346  006201                            ASR     R1              ;GET LINE NUMBER * 1
    111 007350  010160  000002                    MOV     R1,2(R0)        ;SET LINE # IN ARG BLOCK
    112 007354  104375                            EMT     375             ;SEND THE MESSAGE
    113 007356  103006                            BCC     3$              ;BR IF NO ERROR
    114 007360                                    FWARN   #NMSGBF
```

```
     115 007374  000167  000000G          3$:     JMP     RDCMD
     116                                  ;
     117                                  ;  Send message to all terminals.
     118                                  ;
     119 007400  012703  000000G          SNDALL: MOV     #LSTPL,R3       ;GET INDEX # OF LAST LINE
     120 007404  116702  000000G                  MOVB    CORUSR,R2       ;GET OUR LINE INDEX #
     121 007410  016202  000000G                  MOV     LNPRIM(R2),R2   ;GET OUR PRIMARY LINE INDEX #
     122 007414  032763  000000G 000000G 2$:     BIT     #$DILUP,LSW(R3) ;IS THIS LINE LOGGED ON?
     123 007422  001437                           BEQ     1$              ;BRANCH IF NOT
     124 007424  032763  000000G 000000G          BIT     #$DETCH,LSW(R3) ;IS THIS A DETACH LINE?
     125 007432  001033                           BNE     1$              ;BR - DETACH LINES DON'T NEED MESSAGES
     126 007434  020302                           CMP     R3,R2           ;DON'T SEND MESSAGE TO OURSELF
     127 007436  001431                           BEQ     1$
     128 007440  032763  000000G 000000G          BIT     #$TTGAG,LSW7(R3);IS LINE GAGGED?
     129 007446  001407                           BEQ     3$              ;BR IF NOT
     130 007450  105767  000000G                  TSTB    YELEMT          ;Are we YELLing?
     131 007454  001004                           BNE     3$              ;Br if yes -- Override gag
     132 007456  032763  000000G 000000G          BIT     #$INKMN,LSW4(R3);IS JOB IN KMON NOW?
     133 007464  001416                           BEQ     1$              ;IF NOT THEN DON'T SEND MESSAGE
     134 007466  012700  000000G          3$:     MOV     #YELEMT,R0      ;POINT TO EMT ARG BLOCK
     135 007472  010301                           MOV     R3,R1           ;GET LINE INDEX #
     136 007474  006201                           ASR     R1              ;GET LINE NUMBER
     137 007476  010160  000002                   MOV     R1,2(R0)        ;STORE LINE # IN EMT ARG BLOCK
     138 007502  104375                           EMT     375             ;SEND THE MESSAGE
     139 007504  103006                           BCC     1$              ;BR IF NO ERROR
     140 007506                                   FWARN   #NMSGBF         ;REPORT NO MESSAGE BUFFERS
     141 007522  162703  000002           1$:     SUB     #2,R3           ;MOVE ON TO NEXT LINE
     142 007526  001332                           BNE     2$              ;BRANCH IF MORE TO DO
     143 007530  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)
                                ;
007534  010546                  SNDLIN: MOV     R5,-(SP)
                                ;
                                ;  See if a line number was specified
                                ;
007536  004767  0000000G                CALL    SKPSPC          ;Skip over any spaces
007542  121327  000054                  CMPB    (R3),#',        ;Start of line # field?
007546  001407                          BEQ     1$              ;Br if yes
007550  121327  000057                  CMPB    (R3),#'/        ;Start of line # field?
007554  001404                          BEQ     1$              ;Br if yes
007556                                  FABORT  #EM$NLN         ;No line number was specified
                                ;
                                ;  See if we are sending to ALL, OPERATOR, or PARENT
                                ;
007566  005203                  1$:     INC     R3              ;Point past delimiter
007570  004767  0000000G                CALL    SKPSPC          ;Scan up to line number
007574  111300                          MOVB    (R3),R0         ;Get 1st char of "number"
007576  020027  000141                  CMP     R0,#141         ;Is this a lower-case letter?
007602  103402                          BLO     2$              ;Br if not
007604  162700  000040                  SUB     #40,R0          ;Convert lower-case to upper case
007610  120027  000101          2$:     CMPB    R0,#'A          ;Send to ALL?
007614  001003                          BNE     3$              ;Br if not
007616  012701  177777                  MOV     #-1,R1          ;Line # for All
007622  000424                          BR      7$
007624  120027  000117          3$:     CMPB    R0,#'O          ;Send to operator?
007630  001007                          BNE     4$              ;Br if not
007632  116701  0000000G                MOVB    CTRLTT,R1       ;Get operator line #
007636  001016                          BNE     7$
007640                                  FABORT  #NOOPTT         ;No operator terminal
007650  120027  000120          4$:     CMPB    R0,#'P          ;Send to parent job?
007654  001015                          BNE     5$              ;Br if not
007656  116701  0000000G                MOVB    CORUSR,R1       ;Get current job index #
007662  005761  0000000G                TST     LPARNT(R1)      ;Is there a parent job?
007666  001402                          BEQ     7$              ;Br if yes
007670  016101  0000000G                MOV     LPARNT(R1),R1   ;Get # of parent job
007674  112300                  7$:     MOVB    (R3)+,R0        ;Skip over rest of keyword
007676  001407                          BEQ     9$
007700  120027  000040                  CMPB    R0,#'           ;Space?
007704  001373                          BNE     7$
007706  000403                          BR      9$
                                ;
                                ;  We must have an ordinary line number
                                ;
007710  004767  0000000G          5$:     CALL    ACRDEC          ;Accrue line number
007714  006301                          ASL     R1              ;Convert to line index number
                                ;
                                ;  Finished
                                ;
007716  012605                  9$:     MOV     (SP)+,R5
```

SEND command

```
     58 007720  000207                              RETURN
```

```
                                        .SBTTL  DETACH command
                                ;-------------------------------------------------------
                                ;  PROCESS THE DETACH COMMAND
                                ;
007722  004767  000000G         CMDDET: CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACES
007726  032767  000000G 000000G         BIT     #PO$DET,PRIVCO  ;IS USER ALLOWED TO USE DETACHED JOBS?
007734  001004                         BNE     1$              ;BR IF YES
007736                                 FABORT  #EM$DET         ;NOT ALLOWED
                                ;
                                ;  SEE IF ANY SWITCHES WERE SPECIFIED WITH COMMAND
                                ;
007746  121327  000057         1$:     CMPB    (R3),#'/        ;ANY SWITCHES?
007752  001062                         BNE     DETGO           ;BR IF NOT -- START DETACHED JOB
                                ;  ACCRUE AND IDENTIFY SWITCH
007754  005203                         INC     R3              ;SKIP OVER /
007756  012704  000220'               MOV     #DETHD,R4       ;POINT TO TABLE OF SWITCHES
007762  004767  000000G               CALL    SEARCH          ;LOOK UP THE SWITCH
007766  103004                         BCC     2$              ;BR IF FOUND OK
007770                                 FABORT  #INVOPT         ;INVALID SWITCH
                                ;  ACCRUE LINE NUMBER AND SEE IF LEGAL
010000  004767  000000G         2$:     CALL    ACRDEC          ;ACCRUE THE DECIMAL NUMBER
010004  006301                         ASL     R1              ;CONVERT TO LINE INDEX #
010006  020127  000000G               CMP     R1,#FSTDL       ;SEE IF IT IS A DETACHED LINE
010012  103004                         BHIS    3$
010014                         4$:     FABORT  #BDLIN
010024  020127  000000G         3$:     CMP     R1,#LSTDL
010030  101371                         BHI     4$
010032  012700  000000G               MOV     #DETARG,R0      ;POINT TO DETACH EMT ARG BLOCK
010036  006201                         ASR     R1              ;STORE DETACHED JOB #
010040  010167  000002G               MOV     R1,DETARG+2
                                ;  BRANCH OFF TO PROCESSING ROUTINE
010044  000134                         JMP     @(R4)+          ;ENTER PROCESSING ROUTINE
                                ;
                                ;  Process the /CHECK option.
                                ;
010046  112710  000001         DETCHK: MOVB    #1,(R0)         ;SET SUB-FUNCTION CODE FOR EMT
010052  104375                         EMT     375             ;CHECK ON DETACHED JOB
010054  103404                         BCS     DETIDL          ;BR IF JOB FINISHED
010056                                 .PRINT  #RUNMS          ;JOB STILL RUNNING
010064  000403                         BR      DETJMP
010066                         DETIDL: .PRINT  #LINFRE         ;LINE IS IDLE
010074  000167  000000G       DETJMP: JMP     RDCMD           ;FINISHED WITH COMMAND
                                ;
                                ;  Process the /KILL option.
                                ;
010100  112710  000002         DETKIL: MOVB    #2,(R0)         ;SET SUB-FUNCTION CODE FOR EMT
010104  104375                         EMT     375             ;KILL A DETACHED JOB
010106  103767                         BCS     DETIDL          ;BR IF JOB NOT RUNNING
010110                                 .PRINT  #DJABMS         ;PRINT CONFIRMATION
010116  000766                         BR      DETJMP          ;FINISHED
```

```
      1                                 ;
      2                                 ;  Start a new detached job.
      3                                 ;  R3 = Pointer to ASCIZ string with detached job command file name.
      4                                 ;  Try to open the command file to make sure it exists.
      5                                 ;
      6 010120  010302                  DETGO:  MOV     R3,R2           ;Save pointer to start of file name string
      7 010122  012704  000000G                MOV     #R50COM,R4      ;Point to word with default extension (COM)
      8 010126  005005                         CLR     R5              ;Say this is an input file
      9 010130  004767  000000G                CALL    ACRFIL          ;Accrue the file spec
     10 010134  103441                         BCS     10$             ;Br if invalid file spec
     11 010136                                 .LOOKUP #XAREA,#1,#FILNAM ;Try to open the file
     12 010156  103434                         BCS     11$             ;Br if cannot open file
     13 010160                                 .CLOSE  #1              ;Close the file
     14                                 ;
     15                                 ;  Execute EMT to initiate the detached job.
     16                                 ;
     17 010166  012700  000000G                MOV     #DETARG,R0      ;POINT TO DETACH EMT ARG BLOCK
     18 010172  105010                         CLRB    (R0)            ;SET SUB-FUNCTION CODE FOR EMT
     19 010174  010260  000002                 MOV     R2,2(R0)        ;SET ADDRESS OF COMMAND FILE NAME ASCIZ STRING
     20 010200  104375                         EMT     375             ;START A DETACHED JOB
     21 010202  103412                         BCS     1$              ;BR IF NO FREE LINES
     22                                 ;
     23                                 ;  Tell user which line was used
     24                                 ;
     25 010204  010005                         MOV     R0,R5           ;GET DETACHED JOB LINE NUMBER
     26 010206                                 .PRINT  #DLMSG          ;PRINT HEADING
     27 010214  004767  000000G                CALL    PRTDEC          ;DISPLAY LINE #
     28 010220                                 .PRINT  #CRLF           ;TERMINATE PRINT LINE
     29 010226  000722                         BR      DETJMP
     30                                 ;
     31                                 ;  No free lines
     32                                 ;
     33 010230                          1$:    FABORT  #NOFRDL         ;NO FREE LINES
     34                                 ;
     35                                 ;  Invalid file spec
     36                                 ;
     37 010240                          10$:   FABORT  #BDFNAM
     38                                 ;
     39                                 ;  Cannot open file
     40                                 ;
     41 010250                          11$:   FABORT  #EM$FOE
```

```
                                .SBTTL  DATE command
                        ;-------------------------------------------------------------------------
                        ;  Process the DATE command.
                        ;
010260  004767  000000G CMDDAT: CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACES
010264  105713                 TSTB    (R3)            ;DOES HE WANT TO SHOW OR CHANGE THE DATE?
010266  001002                 BNE     3$              ;BR TO SET THE DATE
010270  000167  000000G        JMP     SOPDAT          ;GO DISPLAY THE DATE
                        ;
                        ;  Set a new system date value (operator privilege required).
                        ;
010274  004767  000000G 3$:     CALL    CKPRIV          ;MAKE SURE THIS USER IS PRIVILEGED
                        ;  Get day value.
010300  004767  000000G        CALL    ACRDEC          ;ACCRUE DAY VALUE
010304  005701                 TST     R1              ;CHECK RANGE OF VALUE
010306  003453                 BLE     BADDAT          ;CAN'T BE ZERO
010310  020127  000037         CMP     R1,#31.         ;OR GREATER THAN 31
010314  101050                 BHI     BADDAT
010316  072127  000005         ASH     #5,R1           ;POSITION FOR DATE WORD
010322  010105                 MOV     R1,R5           ;CONSTRUCT DATE WORD IN R5
010324  122327  000055         CMPB    (R3)+,#'-       ;HYPHEN SHOULD BE NEXT
010330  001042                 BNE     BADDAT
                        ;  Get month.
010332  004767  000000G        CALL    GTRD50          ;ACCRUE RAD50 MONTH VALUE
010336  016700  000000G        MOV     R50BUF,R0       ;GET RAD50 VALUE OF MONTH NAME
010342  012701  000001         MOV     #1,R1           ;FIRST MONTH IS # 1
010346  012702  000000G        MOV     #R50MON,R2      ;POINT TO TABLE OF MONTH NAMES
010352  020022         2$:     CMP     R0,(R2)+        ;LOOK UP NAME IN TABLE
010354  001405                 BEQ     1$              ;BR IF FOUND
010356  005201                 INC     R1              ;ADVANCE MONTH NUMBER
010360  020127  000014         CMP     R1,#12.         ;ONLY 12 MONTHS
010364  101772                 BLOS    2$
010366  000423                 BR      BADDAT          ;BAD MONTH NAME
010370  072127  000012 1$:     ASH     #10.,R1         ;POSITION MONTH VALUE
010374  050105                 BIS     R1,R5           ;AND OR INTO DATE WORD
010376  122327  000055         CMPB    (R3)+,#'-       ;SHOULD HAVE ANOTHER HYPHEN
010402  001015                 BNE     BADDAT
                        ;  Get year
010404  004767  000000G        CALL    ACRDEC          ;GET YEAR VALUE
010410  020127  000143         CMP     R1,#99.         ;CHECK FOR REASONABLE UPPER LIMIT
010414  101010                 BHI     BADDAT          ;BR IF TOO HIGH
010416  162701  000110         SUB     #72.,R1         ;YEAR VALUE IS BIASED BY 72
010422  003405                 BLE     BADDAT          ;BR IF TOO SMALL
010424  050105                 BIS     R1,R5           ;FORM DATE WORD
                        ;  Set system date value.
010426  010567  000000G        MOV     R5,SYSDAT       ;SET NEW SYSTEM DATE VALUE
010432  000167  000000G        JMP     RDCMD           ;FINISHED WITH COMMAND
                        ;  Invalid date value.
010436  152767  000010  000000G BADDAT: BISB    #10,INDERR      ;SAVE ERROR LEVEL = SEVERE FOR IND
010444                         FABORT  #INVDAT         ;INVALID DATE
```

```
                                        .SBTTL  TIME command
                                ;-------------------------------------------------------------------------------
                                ;  Process the TIME command.
                                ;
 010454  004767  000000G        CMDTIM: CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACES
 010460  105713                         TSTB    (R3)            ;DOES HE WANT TO SHOW OR CHANGE THE TIME?
 010462  001002                         BNE     2$              ;BR TO SET THE TIME
 010464  000167  000000G                JMP     $OPTIM          ;GO SHOW THE TIME
                                ;
                                ;  Set new system time (requires operator privilege)
                                ;
 010470  004767  000000G        2$:     CALL    CKPRIV          ;MAKE SURE USER HAS OPERATOR PRIVILEGE
                                ;  Get hour value.
 010474  004767  000000G                CALL    ACRDEC          ;ACCRUE HOUR VALUE
 010500  020127  000030                 CMP     R1,#24.         ;ONLY 24 HOUR IN A DAY
 010504  103053                         BHIS    BADTIM
 010506  010104                         MOV     R1,R4
 010510  070427  000074                 MUL     #60.,R4         ;CONVERT TO # MINUTES
 010514  122327  000072                 CMPB    (R3)+,#':       ;COLON SHOULD BE THE DELIMITER
 010520  001045                         BNE     BADTIM
                                ;  Get minute value.
 010522  004767  000000G                CALL    ACRDEC          ;ACCRUE MINUTE VALUE
 010526  020127  000074                 CMP     R1,#60.         ;ONLY 60 MINUTES PER HOUR
 010532  103040                         BHIS    BADTIM
 010534  060105                         ADD     R1,R5           ;COMBINE WITH HOUR VALUE
 010536  005504                         ADC     R4
 010540  012700  000074                 MOV     #60.,R0         ;CONVERT # MINUTES TO # SECONDS
 010544  004767  000000G                CALL    MUL32
                                ;  See if seconds are specified
 010550  122327  000072                 CMPB    (R3)+,#':       ;DELIMITER FOR SECONDS?
 010554  001011                         BNE     3$              ;NO, IGNORE SECONDS
                                ;  Get second value
 010556  004767  000000G                CALL    ACRDEC          ;ACCRUE SECOND VALUE
 010562  020127  000074                 CMP     R1,#60.         ;NO MORE THAN 60 SECONDS PER MINUTE
 010566  103022                         BHIS    BADTIM
 010570  060105                         ADD     R1,R5           ;COMBINE WITH HOURS AND MINUTES
 010572  005504                         ADC     R4
 010574  012700  000074                 MOV     #60.,R0         ;60 HZ CLOCK TICKS PER SECOND
                                ;  Convert time to # clock ticks.
 010600  032767  000000G 000000G 3$:    BIT     #CW$50H,CONFIG  ;50 OR 60 HZ CLOCK?
 010606  001402                         BEQ     1$              ;BR IF 60 HZ
 010610  012700  000062                 MOV     #50.,R0         ;SET FOR 50 HZ CLOCK
 010614  004767  000000G         1$:    CALL    MUL32           ;CONVERT # SECONDS TO # CLOCK TICKS
                                ;  Set new system time.
 010620  010467  000000G                MOV     R4,$SYTIMH      ;HIGH-ORDER TIME VALUE
 010624  010567  000000G                MOV     R5,$SYTIML      ;LOW-ORDER TIME VALUE
 010630  000167  000000G                JMP     RDCMD           ;FINISHED
                                ;  Bad time value entered
 010634  152767  000010  000000G BADTIM: BISB    #10,INDERR      ;SAVE ERROR LEVEL = SEVERE FOR IND
 010642                                 FABORT  #INVTIM         ;INVALID TIME VALUE
```

```
                                        .SBTTL  RESET command
                                ;-------------------------------------------------------------------------
                                ;  The RESET command resets system usage statistics.
                                ;  Operator privilege is required.
                                ;
010652  004767  000000G        CMDRST: CALL    CKSYPV          ;Require SYSPRV privilege
                                ;  Reset data cache statistics
010656  005067  000000G              CLR     DCTRD           ;TOTAL NUMBER OF READS FROM SHARED FILES
010662  005067  000000G              CLR     DCCRD           ;NUMBER OF READS FOUND IN CACHE
010666  005067  000000G              CLR     DCTWR           ;TOTAL NUMBER OF WRITES TO SHARED FILES
010672  005067  000000G              CLR     DCCWR           ;NUMBER OF WRITES THAT UPDATE CACHE DATA
                                ;  Reset 32-bit data cells
010676  012701  010720'             MOV     #RSTVEC,R1      ;POINT TO VECTOR OF ADDRESSES TO CLEAR
010702  012102                  1$: MOV     (R1)+,R2        ;GET ADDRESS OF A CELL TO ZERO
010704  001403                      BEQ     2$              ;BR IF HIT END OF LIST
010706  005022                      CLR     (R2)+           ;CLEAR HIGH-ORDER WORD
010710  005012                      CLR     (R2)            ;CLEAR LOW-ORDER WORD
010712  000773                      BR      1$
010714  000167  000000G          2$: JMP     RDCMD           ;FINISHED
                                ;
                                ;  Vector of 32-bit cells to clear.
                                ;
010720  000000G                 RSTVEC: .WORD   TMTOTH
010722  000000G                         .WORD   TMUSRH
010724  000000G                         .WORD   TMSWTH
010726  000000G                         .WORD   TMIOH
010730  000000G                         .WORD   TMSWPH
010732  000000G                         .WORD   TMIOWH
010734  000000G                         .WORD   TMIDLH
010736  000000G                         .WORD   CASTRO
010740  000000G                         .WORD   CASTBR
010742  000000G                         .WORD   CASCBR
010744  000000G                         .WORD   CASTWO
010746  000000G                         .WORD   CASTBW
010750  000000G                         .WORD   CASCUP
010752  000000                          .WORD   0
```

```
                                        .SBTTL  OFF command
                                ;-------------------------------------------------------
                                ;  LOG OFF.
                                ;
 010754  116701  000000G       CMDOFF: MOVB    CORUSR,R1       ;GET USER INDEX #
                                ;
                                ;  See if any qualifiers were specified with command
                                ;
 010760  105067  167061                 CLRB    OFFNWF          ;Clear NOWARN flag
 010764  012704  000250'                MOV     #OFFHD,R4       ;Point to parsing table
 010770  004767  000000G                CALL    OPTLST          ;Process any qualifiers
                                ;
                                ;  See if this is a primary process logging off with active subprocesses
                                ;
 010774  020127  000000G                CMP     R1,#LSTPL       ;Is this a primary process?
 011000  101027                         BHI     14$             ;Br if not
 011002  105767  167037                 TSTB    OFFNWF          ;Was /NOWARN qualifier specified?
 011006  001024                         BNE     14$             ;Br if yes -- Don't worry about subprocesses
 011010  105767  000000G              TSTB    STPFLG          ;Is this a STOP request?
 011014  001021                         BNE     14$             ;Br if yes -- Don't worry about subprocesses
 011016  032761  000000G 000000G      BIT     #$PRGLK,LSW5(R1);Did we exit a locked program?
 011024  001015                         BNE     14$             ;Br if yes -- Don't worry about subprocesses
 011026  032761  000000G 000000G      BIT     #$NOIN,LSW3(R1) ;Are we allowed to have terminal input?
 011034  001011                         BNE     14$             ;Br if not
 011036  032761  000000G 000000G      BIT     #$LOFCF,LSW9(R1);Have we already done a logAoff command file?
 011044  001005                         BNE     14$             ;Do logoff if so
 011046  004767  001142                 CALL    CKOFSP          ;See if we need to warn about subprocesses
 011052  103002                         BCC     14$             ;Br if we should logoff
 011054  000167  000000G              JMP     RDCMD           ;Abort the logoff
                                ;
                                ;  Do the logoff.
                                ;
 011060  052761  000000G 000000G 14$:  BIS     #$NOIN,LSW3(R1) ;DON'T ALLOW ABORT OF LOGOFF
 011066  005067  000000G              CLR     RESDEV          ;RELEASE ALL DEVICE/FILE ACCESS RESTRICTIONS
                                ;
                                ;  Close all of user's files.
                                ;
 011072  004767  000000G                CALL    PRGALL          ;PURGE ALL OF USER'S CHANNELS
 011076  004767  000000G                CALL    INDABT          ;Abort IND and nested command files
 011102  004767  000000G                CALL    ABRTCF          ;CLOSE THE CONTROL FILE
 011106  052761  000000G 000000G      BIS     #$NOIN,LSW3(R1) ;CLEARED BY ABRTCF(POPCF), SET IT AGAIN
                                ;
                                ;  Determine if we need to invoke a logoff command file
                                ;
 011114  005767  000000G                TST     LOFSPC          ;Is there a logoff command file?
 011120  001471                         BEQ     3$              ;Br if not
 011122  004767  000000G                CALL    PUSHCF          ;Prepare to open new command file
 011126  112767  000001  000000G      MOVB    #1,SERFLG       ;Do .SERR
 011134                                 .LOOKUP #XAREA,#CFCHAN,#LOFSPC ;Try to open the command file
 011154  112767  000000  000000G      MOVB    #0,SERFLG       ;Do .HERR but don't affect carry flag
 011162  103013                         BCC     10$             ;Br if open was ok
 011164  004767  000000G        11$:    CALL    POPCF           ;Pop command file status
 011170  005067  000000G                CLR     LOFSPC          ;No logoff command file
 011174                                 FERR    #EM$OLO         ;Cannot open logoff command file
 011210  000435                         BR      3$
 011212                         10$:    .READW  #XAREA,#CFCHAN,#CFBUF,#256.,#0 ;Read 1st block of file
 011250  103745                         BCS     11$             ;Br if read error
```

```
     58 011252  052761  000000G 000000G          BIS     #$CFOPN,LSW4(R1);Say CFCHAN is open
     59 011260  052761  000000G 000000G          BIS     #$LOFCF,LSW9(R1);Say we are processing logoff command file
     60 011266  042761  000000G 000000G          BIC     #$DOOFF,LSW(R1) ;Say not doing logoff processing yet
     61 011274  005067  000000G                 CLR     LOFSPC          ;Say logoff command file has been started
     62 011300  000167  000000G                 JMP     RDCMD           ;Go process 1st command in command file
     63                                     ;
     64                                     ;  Enable privilege during logoff processing
     65                                     ;
     66 011304  012702  000000G          3$:    MOV     #PRIVC0,R2      ;Point to current privilege word
     67 011310  012703  000000G                 MOV     #PRIVS0,R3      ;Point to set privileges
     68 011314  012700  000000G                 MOV     #PVNPW,R0       ;Get # privilege words
     69 011320  012722  177777          12$:   MOV     #177777,(R2)+   ;Set all privilege flags
     70 011324  012723  177777                 MOV     #177777,(R3)+
     71 011330  077005                          SOB     R0,12$
     72 011332  052767  000000G 000000G        BIS     #LF$WRT,LOGFLG  ;ENABLE WRITES TO LOG FILE
     73                                     ;
     74                                     ;  Dismount all devices that were mounted by this job.
     75                                     ;
     76 011340  004767  000000G                 CALL    DMTALL          ;Dismount all mounted devices
     77                                     ;
     78                                     ;  Print usage statistics for line
     79                                     ;
     80 011344  032761  000000G 000000G        BIT     #$VNOTT,LSW(R1) ;IS THIS JOB CONNECTED TO A TERMINAL?
     81 011352  001033                          BNE     7$              ;BR IF NOT
     82 011354  016100  000000G                 MOV     LNPRIM(R1),R0   ;GET PRIMARY LINE #
     83 011360  042760  000000G 000000G        BIC     #$CTRLS,LSW3(R0);MAKE SURE OUTPUT IS ENABLED
     84 011366  020127  000000G                 CMP     R1,#LSTDL       ;Is this a subprocess?
     85 011372  101403                          BLOS    13$             ;Br if not
     86 011374  005760  000000G                 TST     LWINDO(R0)      ;Does primary process have windowing on?
     87 011400  001020                          BNE     7$              ;Br if yes -- Don't display logoff time
     88 011402                          13$:    .PRINT  #CRLF
     89 011410  105767  000000G                 TSTB    STPFLG          ;IS SYSTEM BEING SHUT DOWN?
     90 011414  001403                          BEQ     5$              ;BR IF NOT
     91 011416                                  .PRINT  #SHTMSG         ;PRINT SHUT-DOWN MESSAGE
     92 011424  004767  000000G          5$:    CALL    PRTTIM          ;PRINT USAGE INFO
     93 011430                                  .TTYOUT #LF             ;PUT OUT EXTRA LF
     94 011440  000420                          BR      TIMUP
     95                                     ;  IF PRTTIM DIDN'T DO CPU CALCULATION WE MUST DO IT HERE.
     96 011442  016104  000000G          7$:    MOV     LCPUHI(R1),R4   ;GET HIGH-ORDER CPU TIME (CLOCK TICKS)
     97 011446  016105  000000G                 MOV     LCPULO(R1),R5   ;GET LOW-ORDER CPU TIME (CLOCK TICKS)
     98 011452  016703  000000G                 MOV     TK1SEC,R3       ;GET # CLOCK TICKS PER SECOND
     99 011456  004767  000000G                 CALL    DIVIDE          ;CONVERT TIME VALUE TO # SECONDS
    100 011462  012700  000012                  MOV     #10.,R0         ;CONVERT TIME VALUE TO 0.1 SECOND
    101 011466  004767  000000G                 CALL    MUL32           ;UNITS BY MULTIPLYING BY 10
    102 011472  010467  000000G                 MOV     R4,CPUAH        ;SAVE HIGH-ORDER PART (0.1 SEC)
    103 011476  010567  000000G                 MOV     R5,CPUAL        ;SAVE LOW-ORDER PART (0.1 SEC)
    104                                     ;
    105                                     ;  If user logged on, then accumulate his connect time
    106                                     ;
    107 011502  016105  000000G          TIMUP: MOV     LPROJ(R1),R5    ;DO WE HAVE A PPN FOR USER?
    108 011506  001535                          BEQ     NOTIM           ;BR IF NOT -- DIDN'T LOG ON
    109                                     ;  DO A DEASSIGN
    110 011510  012702  000000G                 MOV     #ASNTBL,R2      ;CLEAR ALL OF HIS ASSIGNS
    111 011514  005022                   1$:    CLR     (R2)+
    112 011516  020227  000000G                 CMP     R2,#ASNEND
    113 011522  103774                          BLO     1$
    114                                     ;  Try to open accounting file
```

```
     115 011524                                         .LOOKUP #XAREA,#1,#AUTHFN
     116 011544  103004                                 BCC     2$                  ;BR IF OPEN OK
     117 011546                                         .PRINT  #COAF               ;PRINT WARNING MESSAGE
     118 011554  000512                                 BR      NOTIM
     119                                        ;  Search file for user's ppn entry
     120 011556  005002                         2$:     CLR     R2                  ;SET BLOCK # TO 0
     121 011560  016103  000000G                        MOV     LPROG(R1),R3        ;GET PROGRAMMER #
     122 011564                                 6$:     .READW  #XAREA,#1,#BLKO,#256.,R2
     123 011622  103464                                 BCS     3$                  ;BR IF END OF FILE HIT
     124 011624  004767  000544                         CALL    CVTBUF              ;COMPLEMENT CONTENTS OF BUFFER
     125 011630  012704  000000G                        MOV     #BLKO,R4            ;SEARCH FOR RECORD IN BLOCK
     126 011634  012700  000000G                        MOV     #ARNRPB,R0          ;# RECORDS PER BLOCK
     127 011640  020564  000000G                5$:     CMP     R5,AR$PRJ(R4)       ;DO PROJECT #'S MATCH?
     128 011644  001003                                 BNE     8$                  ;BR IF NOT
     129 011646  020364  000000G                        CMP     R3,AR$PRG(R4)       ;HOW ABOUT PROGRAMMER NUMBERS
     130 011652  001406                                 BEQ     4$                  ;BR IF WE FOUND THE RECORD
     131 011654  062704  000000G                8$:     ADD     #AR$$SZ,R4          ;KEEP LOOKING IN BLOCK
     132 011660  005300                                 DEC     R0                  ;ANY MORE RECORDS IN THIS BLOCK?
     133 011662  001366                                 BNE     5$                  ;BR IF YES
     134 011664  005202                                 INC     R2                  ;GO READ NEXT BLOCK
     135 011666  000736                                 BR      6$
     136                                        ;  Found record.  Add in new connect time
     137 011670  016700  000000G                4$:     MOV     MINTIM,R0           ;GET CURRENT TIME
     138 011674  166100  000000G                        SUB     LCONTM(R1),R0       ;CALCULATE CONNECT TIME
     139 011700  005200                                 INC     R0
     140 011702  060064  000000G                7$:     ADD     R0,AR$CON(R4)       ;ACCUMULATE CONNECT TIME
     141                                        ;  COUNT NUMBER OF SESSIONS
     142 011706  005264  000000G                        INC     AR$CNT(R4)          ;INC SESSION COUNTER
     143                                        ;  ACCUMULATE CPU TIME
     144 011712  066764  000000G 000000G               ADD     CPUAL,AR$CPL(R4)        ;ADD IN LOW-ORDER CPU TIME
     145 011720  005564  000000G                        ADC     AR$CPH(R4)          ;ADD CARRY
     146 011724  066764  000000G 000000G               ADD     CPUAH,AR$CPH(R4)        ;ADD IN HIGH-ORDER PART
     147                                        ;  WRITE UPDATED ACCOUNTING RECORD BACK TO FILE
     148 011732  004767  000436                         CALL    CVTBUF              ;COMPLEMENT CONTENTS OF BUFFER
     149 011736                                         .WRITW  #XAREA,#1,#BLKO,#256.,R2
     150 011774                                 3$:     .CLOSE  #1
     151                                        ;
     152                                        ;  Close job's log file
     153                                        ;
     154 012002  004767  000000G                NOTIM:  CALL    LOGCLS              ;CLOSE LOG FILE
     155                                        ;
     156                                        ;  Broadcast message to any monitoring jobs telling them we are logging off
     157                                        ;
     158 012006  012700  000000G                        MOV     #GENMON,R0          ;Point to emt argument block
     159 012012  012760  000000G 000002                MOV     #JS$OFF,2(R0)       ;Set status code
     160 012020  104375                                 EMT     375                 ;Broadcast status message
     161                                        ;
     162                                        ;  Enter TSX to finish log off.
     163                                        ;
     164 012022  105767  000000G                        TSTB    STPFLG              ;ARE WE DOING A SYSTEM SHUTDOWN?
     165 012026  001460                                 BEQ     1$                  ;BR IF NOT
     166 012030  003046                                 BGT     15$                 ;Br if $STOP or BOOT specified
     167                                        ;
     168                                        ;  Shutdown was specified.
     169                                        ;
     170 012032  052761  000000G 000000G                BIS     #$NOABT,LSW9(R1);Don't allow job abort
     171 012040  120127  000000G                        CMPB    R1,#LSTPL           ;Is this a detached job?
```

```
     172 012044  103403                          BLO     10$             ;Br if not detached
     173 012046  120127  000000G                 CMPB    R1,#LSTDL       ;In the detached range?
     174 012052  101435                          BLOS    15$             ;Yes, job is detached
     175 012054  126727  000000G 000001  10$:    CMPB    PVON,#1         ;Are we the last primary/virtual job?
     176 012062  003031                          BGT     15$             ;Br if more jobs exist
     177                                 ;
     178                                 ;  Kill all jobs (only detached jobs should be active).
     179                                 ;
     180 012064  012702  000000G                 MOV     #LSTSL,R2       ;GET # OF LAST LINE
     181 012070  120267  000000G         11$:    CMPB    R2,CORUSR       ;IS THIS OUR LINE?
     182 012074  001405                          BEQ     12$             ;BR IF YES
     183 012076  012700  000000G                 MOV     #KILEMT,R0      ;POINT TO KILL EMT ARG BLOCK
     184 012102  010260  000004                  MOV     R2,4(R0)        ;SET # OF LINE TO KILL
     185 012106  104375                          EMT     375             ;KILL THE LINE
     186 012110  162702  000002          12$:    SUB     #2,R2           ;DO NEXT LINE
     187 012114  001365                          BNE     11$
     188                                 ;
     189                                 ;  Determine if the spooler is still active.
     190                                 ;
     191 012116  004767  000000G         13$:    CALL    SPLACT          ;Is the spooler still active?
     192 012122  103011                          BCC     15$             ;Br if spooler is not active.
     193 012124                                  .TWAIT  #XAREA,#TIMSPL  ;Wait 2 seconds -- check again
     194 012144  000764                          BR      13$
     195                                 ;
     196                                 ;  $STOP or BOOT issued.
     197                                 ;
     198 012146  042761  000000G 000000G 15$:    BIC     #$NOABT,LSW9(R1);Allow job abort (necessary for log off)
     199 012154  126727  000000G 000001         CMPB    TOTON,#1        ;Are we the last job on system?
     200 012162  003002                          BGT     1$              ;Br if more jobs exist
     201 012164  000167  000000G                 JMP     DOSTOP          ;GO STOP THE SYSTEM
     202 012170  005061  000000G          1$:    CLR     LPROJ(R1)       ;CLEAR PROJECT/PROGRAMMER #
     203 012174  005061  000000G                 CLR     LPROG(R1)
     204 012200  012700  000000G                 MOV     #OFFEMT,R0      ;POINT TO LOG-OFF EMT ARG BLOCK
     205 012204  104375                          EMT     375             ;LOG OFF THIS JOB
```

```
                                ;-------------------------------------------------------------------------------
                                ;  Subroutine called if the /NOWARN qualifier is specified with the OFF
                                ;  command.
                                ;
012206  105267  165633          OFFNWN: INCB    OFFNWF          ;Set NOWARN flag
012212  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.
                                ;
012214  010246                  CKOFSP: MOV     R2,-(SP)
012216  010546                          MOV     R5,-(SP)
                                ;
                                ;  See if this process has any active subprocesses
                                ;
012220  005005                          CLR     R5              ;Count active subprocesses in R5
012222  016102  000000G                 MOV     LSECPT(R1),R2   ;Point to table of subprocess #'s
012226  012700  000000G                 MOV     #MAXSEC,R0      ;Get max # subprocesses
012232  001454                          BEQ     7$              ;Br if none to check
012234  105722                  1$:     TSTB    (R2)+           ;Is this subprocess active?
012236  001401                          BEQ     4$              ;Br if not
012240  005205                          INC     R5              ;Count # active subprocesses
012242  077004                  4$:     SOB     R0,1$           ;Loop to check all
012244  005705                          TST     R5              ;Any active subprocesses?
012246  001446                          BEQ     7$              ;Br if not
                                ;
                                ;  There are active subprocesses.
                                ;  Print warning message.
                                ;
012250  020527  000001                  CMP     R5,#1           ;One active subprocess?
012254  003004                          BGT     2$              ;Br if more than one
012256                                  .PRINT  #TM$SA1         ;You have 1 active subprocess
012264  000410                          BR      5$
012266                          2$:     .PRINT  #TM$SA3         ;You have
012274  004767  000000G                 CALL    PRTDEC          ;Print # subprocesses
012300                                  .PRINT  #TM$SA4         ;active subprocesses
                                ;
                                ;  Read response
                                ;
012306                          5$:     .GTLIN  #BLK0,#TM$SA2   ;Print prompt and read response
                                ;
                                ;  If first non-blank letter of response is "Y" then logoff,
                                ;  otherwise abort the logoff.
                                ;
012326  012702  000000G                 MOV     #BLK0,R2        ;Point to response buffer
012332  112200                  3$:     MOVB    (R2)+,R0        ;Get next char of response
012334  001411                          BEQ     8$              ;Br if hit end of line
012336  120027  000040                  CMPB    R0,#40          ;Ignore spaces
012342  001773                          BEQ     3$
```

```
      58 012344  120027  000131                    CMPB    R0,#'Y          ;Is response Yes?
      59 012350  001405                            BEQ     7$              ;Br if yes
      60 012352  120027  000171                    CMPB    R0,#'y          ;Allow lower-case too
      61 012356  001402                            BEQ     7$
      62                                ;
      63                                ;  Abort the logoff
      64                                ;
      65 012360  000261                 8$:        SEC                     ;Signal abort on return
      66 012362  000401                            BR      9$
      67                                ;
      68                                ;  Allow the logoff
      69                                ;
      70 012364  000241                 7$:        CLC                     ;Signal to logoff
      71                                ;
      72                                ;  Finished
      73                                ;
      74 012366  012605                 9$:        MOV     (SP)+,R5
      75 012370  012602                            MOV     (SP)+,R2
      76 012372  000207                            RETURN
```

```
                                ;-------------------------------------------------------------------------------
                                ;  Complement the contents of BLKO buffer.
                                ;  We do this to mildly encript the authorization file.
                                ;
012374  010146                CVTBUF: MOV     R1,-(SP)
012376  010246                        MOV     R2,-(SP)
012400  012701  0000000G              MOV     #BLKO,R1        ;POINT TO BUFFER
012404  012702  000400                MOV     #256.,R2        ;GET # WORDS TO COMPLEMENT
012410  005121                1$:     COM     (R1)+           ;COMPLEMENT CONTENTS OF BUFFER
012412  077202                        SOB     R2,1$
012414  012602                        MOV     (SP)+,R2
012416  012601                        MOV     (SP)+,R1
012420  000207                        RETURN
```

```
                                        .SBTTL  KILL command
                                ;-------------------------------------------------------------------------------
                                ;  The KILL command is used to abort the specified job.
                                ;
012422  012705  000000G         CMDKIL: MOV     #KILEMT,R5      ;Point to Kill EMT arg block
012426  000405                         BR      JOBCOM
                                
                                        .SBTTL  SUSPEND command
                                ;-------------------------------------------------------------------------------
                                ;  Suspend the execution of a specified job
                                ;
012430  012705  000000G         CMDSPN: MOV     #SJEMT,R5       ;Point to suspend EMT arg block
012434  000402                         BR      JOBCOM
                                
                                        .SBTTL  RESUME command
                                ;-------------------------------------------------------------------------------
                                ;  Resume the execution of a suspended job
                                ;
012436  012705  000000G         CMDRSM: MOV     #RJEMT,R5       ;Point to resume EMT arg block
                                ;
                                ;  Accrue the job number
                                ;
012442  004767  000000G         JOBCOM: CALL    CVTTAB          ;convert tab and FF chars to spaces
012446  004767  000000G                 CALL    ACRDEC          ;accrue decimal value
012452  006301                         ASL     R1              ;Convert # to job index number
012454  001403                         BEQ     1$              ;Br if invalid line number
012456  020127  000000G                CMP     R1,#LSTSL       ;Is this a valid line number
012462  101404                         BLOS    2$              ;Br if ok
012464                          1$:     FABORT  #BDLIN          ;Invalid line number
                                ;
                                ;  See if we are privileged to affect this job
                                ;
012474  010102                  2$:     MOV     R1,R2           ;Get job # to R2 for CKACOJ
012476  004767  000000G                 CALL    CKACOJ          ;Can we access this job
012502  103404                         BCS     9$              ;Br if not
                                ;
                                ;  Execute EMT to affect the job
                                ;
012504  010500                         MOV     R5,R0           ;Point to EMT arg list
012506  010160  000004                 MOV     R1,4(R0)        ;Set job number
012512  104375                         EMT     375             ;Perform the function
                                ;
                                ;  Finished
                                ;
012514  000167  000000G         9$:     JMP     RDCMD
```

```
                                        .SBTTL  BOOT and $STOP commands
                                ;-------------------------------------------------------------------------
                                ;  Reboot RT-11.
                                ;
012520  004767  000000G         CMDBOT: CALL    CKPRIV          ;USER MUST HAVE OPERATOR COMMAND PRIVILEGE
012524  004767  000000G                 CALL    CVTTAB          ;CONVERT TAB AND FF CHARS TO SPACES
                                ;
                                ;  See if a boot device was specified.
                                ;
012530  005067  000000G                 CLR     BOTDEV          ;ASSUME NO BOOT DEVICE WILL BE SPECIFIED
012534  105713                         TSTB    (R3)            ;WAS A BOOT DEVICE SPECIFIED?
012536  001450                         BEQ     7$              ;BR IF NOT -- USE SYSTEM DEVICE
                                ;  Accrue device name.
012540  004767  000000G                 CALL    GTRD50          ;ACCRUE RAD50 DEVICE NAME
012544  016767  000000G 000000G         MOV     R50BUF,FILNAM   ;SAVE RAD50 NAME OF BOOT DEVICE
012552  005067  000002G                 CLR     FILNAM+2        ;Clear remaining file specification
012556  005067  000004G                 CLR     FILNAM+4
012562  005067  000006G                 CLR     FILNAM+6
012566                                  .LOOKUP #XAREA,#1,#FILNAM;Open a channel to the boot device
012606  103510                         BCS     NODEV           ;Br if lookup is unsuccessful
012610                                  .CSTAT  #XAREA,#1,#CINFO;Obtain channel information from boot device
012630  016767  165162  000000G         MOV     CINFO+12,BOTDEV ;Transfer RAD50 device name to boot device
012636  066767  165152  000000G         ADD     CINFO+10,BOTDEV ;Add unit number to boot device name
012644  062767  000036  000000G         ADD     #36,BOTDEV      ;Convert unit number to rad50
012652                                  .CLOSE  #1              ;Close channel to boot device
                                ;
                                ;  Make sure system is idle.
                                ;
                                ;  See if any other users are logged on.
012660  005005                 7$:     CLR     R5              ;IDLE FLAG
012662  126727  000000G 000001         CMPB    NUMON,#1        ;ARE WE ONLY JOB LOGGED ON?
012670  003404                         BLE     1$              ;BR IF YES
012672                                  .PRINT  #OTHRON         ;OTHER USERS LOGGED ON
012700  005205                         INC     R5              ;REMEMBER NOT IDLE
                                ;  See if the spooler is idle.
012702  004767  000000G         1$:     CALL    SPLACT          ;SEE IF SPOOLER IS ACTIVE
012706  103004                         BCC     2$              ;BR IF SPOOLER IS IDLE
012710                                  .PRINT  #SPLPND         ;THERE ARE PENDING SPOOL FILES
012716  005205                         INC     R5              ;REMEMBER SYSTEM IS NOT IDLE
                                ;  See if we need to ask for confirmation.
012720  005705                 2$:     TST     R5              ;WAS SYSTEM IDLE?
012722  001420                         BEQ     10$             ;BR IF YES
012724                                  .PRINT  #STPASK         ;ASK FOR CONFIRMATION
012732                          5$:     .TTYIN                  ;ACCEPT INPUT LINE
012736  120027  000131                 CMPB    R0,#'Y          ;DID HE SAY YES?
012742  001410                         BEQ     10$             ;BR IF YES
012744  120027  000171                 CMPB    R0,#171         ;LOWER-CASE 'Y'
012750  001405                         BEQ     10$
012752  120027  000012                 CMPB    R0,#LF          ;EAT REST OF LINE
012756  001365                         BNE     5$
012760  000167  000000G                 JMP     RDCMD           ;GO GET NEXT COMMAND
                                ;
                                ;  Force logoff of all users.
                                ;
012764  112767  000001  000000G 10$:     MOVB    #1,STPFLG       ;SET FLAG SAYING SYSTEM IS BEING STOPPED
012772  012702  000000G                 MOV     #LSTSL,R2       ;GET # OF LAST LINE
012776  120267  000000G         11$:    CMPB    R2,CORUSR       ;IS THIS OUR LINE?
```

```
     58 013002  001405                        BEQ     12$             ;BR IF YES
     59 013004  012700  000000G               MOV     #KILEMT,R0      ;POINT TO KILL EMT ARG BLOCK
     60 013010  010260  000004                MOV     R2,4(R0)        ;SET # OF LINE TO KILL
     61 013014  104375                        EMT     375             ;KILL THE LINE
     62 013016  162702  000002          12$:  SUB     #2,R2           ;DO NEXT LINE
     63 013022  001365                        BNE     11$
     64                                 ;
     65                                 ;  Now enter logoff processing for our line.
     66                                 ;  It will jump to DOSTOP after it finishes.
     67                                 ;
     68 013024  000167  175724                JMP     CMDOFF          ;LOG OFF OUR LINE
     69                                 ;
     70                                 ;  Invalid boot device - could not locate boot device
     71                                 ;
     72 013030                          NODEV: FABORT  #BADBOT         ;Say bad boot device
     73
```

```
      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 013040  004767  000000G        CMDSHT: CALL    CKPRIV          ;USER MUST HAVE OPERATOR PRIV
      7 013044  112767  177777  000000G       MOVB    #-1,STPFLG      ;SAY SYSTEM SHUTDOWN TAKING PLACE
      8 013052  005067  000000G               CLR     BOTDEV          ;REBOOT FROM SYSTEM DISK
      9 013056  000167  000000G               JMP     RDCMD
     10            000001                    .END
Errors detected:  0

*** Assembler statistics


Work  file  reads:  0
Work  file writes:  0
Size of work file:  10920 Words  ( 43 Pages)
Size of core pool:  18176 Words  ( 71 Pages)
Operating  system:  RT-11

Elapsed time:  00:01:50.40
,LP:TSKM2A=DK:TSKM2A/C/N:SYM
```

Cross reference table (CREF V05.05)

```
$CFABT    1-80      7-17      7-34
$CFOPN    1-90     40-58
$CTRLS    1-71     40-83
$DETCH    1-69     33-36     33-124
$DILUP    1-81     33-97     33-122
$DOOFF    1-87     40-60
$INKMN    1-79     33-103    33-132
$LOFCF    1-145    40-25     40-59
$NOABT    1-148    40-170    40-198
$NOIN     1-63     40-23     40-33     40-41
$PRGLK    1-67     40-21
$QUIET    1-96      7-21
$SCCA     1-39      7-15
$TTGAG    1-90     33-101    33-128
$VNOTT    1-48     40-80
...V1     7-23      7-27      7-50      8-63     11-71     12-53     14-16     14-24     14-37     22-50     23-91     25-54
         28-47     30-33     31-11     31-18     31-18     31-18     31-20     31-21     31-21     31-27     31-27     31-32
         31-32     31-34     31-34     31-38     31-38     31-39     31-39     32-6      32-10     32-12     32-16     32-20
         32-22     32-27     32-33     32-36     32-39     32-41     32-44     32-46     32-50     32-52     32-53     33-99
         33-107    35-39     35-41     35-49     36-11     36-11     36-11     36-13     36-13     36-26     36-28     40-49
         40-49     40-49     40-56     40-56     40-88     40-91     40-93     40-115    40-115    40-115    40-117    40-122
         40-122    40-149    40-149    40-150    40-150    40-193    41-40     41-42     41-44     41-48     44-19     44-19
         44-19     44-21     44-25     44-25     44-33     44-38     44-43
...V2    23-91     23-91     23-91#    23-91#    31-18     31-18     31-18#    31-18#    31-21     31-21#    31-27     31-27
         31-27#    31-27#    31-32     31-32     31-32#    31-32#    31-34     31-34     31-34#    31-34#    31-38     31-38#
         31-39     31-39#    36-11     36-11     36-11#    36-11#    36-13     36-13#    40-49     40-49     40-49#    40-49#
         40-56     40-56     40-56#    40-56#    40-115    40-115    40-115#   40-115#   40-122    40-122    40-122#   40-122#
         40-149    40-149    40-149#   40-149#   40-150    40-150#   44-19     44-19     44-19#    44-19#    44-21     44-21
         44-21#    44-21#    44-25     44-25#
ABRTCF    1-115    40-40
ACRDEC    1-122    28-21     30-19     31-84     34-52     35-21     37-14     37-39     38-14     38-22     38-33     43-24
ACRFIL    1-121    18-32     23-34     36-9
ACROCT    1-124    28-11     28-15
AF$BYA    1-61      5-6
AF$CCA    1-39      5-19
AF$DBG    1-43      5-8
AF$DUP    1-40      5-7
AF$HIE    1-146     5-9
AF$IND    1-40      5-10
AF$IOP    1-147     5-11      5-12
AF$MEM    1-61      5-14
AF$NOI    1-146     5-15
AF$NOW    1-146     5-16
AF$NPW    1-39      5-17
AF$PLK    1-43      5-13
AF$SCA    1-64      5-21
AF$SET    1-40      5-20
AF$TPO    1-42      5-22
AF$UCL    1-40      5-23
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    1-132    31-17*    31-18
ALDEMT    1-68     11-49
ALDEX     1-131    31-11     31-11
ALFN      1-141    31-27
AMBOPT    1-122    29-66
```

Cross reference table (CREF V05.05)

```
AR$$SZ    1-144    40-131
AR$CNT    1-143    40-142*
AR$CON    1-143    40-140*
AR$CPH    1-143    40-145*   40-146*
AR$CPL    1-143    40-144*
AR$PRG    1-143    40-129
AR$PRJ    1-143    40-127
ARNRPB    1-144    40-126
ASCIS     8-53      8-62#
ASCSOK    8-65      8-81#
ASDELM    8-17      9-7#
ASDEX     1-119     8-63
ASFID     8-36      8-57#
ASKLNM    1-118     8-8*      8-52*     8-85
ASNEND    1-83     10-14     10-28     40-112
ASNOVF    1-119     8-97
ASNSRC    1-103     8-82      8-90      8-94      25-31     29-21
ASNTBL    1-81     10-12     10-22     40-110
ASSMPL    8-42#
AT$$SZ    1-82     10-27
AT$DEV    1-82      8-84      8-105*   10-26*    25-33     29-23
AT$EXT    1-82      8-108*
AT$FIL    1-82      8-106*    8-107*
AT$LOG    1-82      8-102*   10-23     10-25*
AT$SIZ    1-82      8-103*
AUTHFN    1-136    40-115
BADBOT    1-148    44-72
BADCMD    1-118     8-15     29-12
BADDAT   37-16     37-18     37-22     37-33     37-37     37-41     37-43     37-49#
BADTIM   38-16     38-20     38-24     38-35     38-49#
BDFNAM    1-121    18-99     23-36     24-17     36-37
BDLIN     1-134    33-34     35-25     43-29
BELL      2-7#     33-49
BLANK     2-6#     33-65
BLKO      1-117    31-32     31-34     40-122    40-125    40-149    41-48     41-53     42-7
BLKWDS    2-10#
BOTDEV    1-78     44-10*    44-22*    44-23*    44-24*    45-8*
C.CSW     1-88     27-27
C.DEVQ    1-88     27-31
CASCBR    1-75     39-32
CASCUP    1-75     39-35
CASTBR    1-75     39-31
CASTBW    1-75     39-34
CASTRO    1-76     39-30
CASTWO    1-76     39-33
CCLSAV    1-73     27-59
CD$$SZ    1-89    26-115
CD$BAS    1-89     26-96
CD$DVU    1-89     26-94
CDBUF     1-47     26-94     26-96
CDGET     1-47     26-93
CFBUF     1-73     40-56
CFCHAN    1-95     40-49     40-49     40-56     40-56
CFPNT     1-96      7-8
CFSPND    1-100     7-8*      7-35      7-37*
CFSTOP    1-41      7-10
```

Cross reference table (CREF V05.05)

```
CFSTRT    1-41      7-36
CHKALC    1-62     26-69
CHKDEV    1-139    13-30     18-53     26-75     27-18
CHKDLM    1-130    29-50
CHKMNT    1-120    25-50     26-48
CHKMTX    1-120    25-52     26-54     26-106
CINFO     4-7#     44-21     44-22     44-23
CKACOJ    1-44     43-34
CKOFSP   40-27     41-19#
CKPRIV    1-122    31-5      31-44     31-50     31-78     31-90     32-59     32-65     32-72     32-81     32-89     32-95
         32-101    32-107    33-6      37-12     38-12     44-5      45-6
CKSYDV   26-42     27-11#
CKSYPV    1-44     16-4      17-4      39-6
CLRPRV    1-46     16-6
CMDALC    1-31     11-7#
CMDASN    1-30      8-7#
CMDBOT    1-33     44-5#
CMDBUF    1-114     7-23
CMDCCL    1-139    26-127
CMDDAT    1-29     37-5#
CMDDET    1-32     35-5#
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     32-114#
CMDINI    1-32     26-8#
CMDINS    1-28     15-5#
CMDKIL    1-31     43-5#
CMDMNT    1-30     22-8#
CMDMON    1-30     28-7#
CMDOFF    1-26     40-5#     44-68
CMDPAU    1-32      7-8#
CMDREM    1-31     13-8#
CMDRSM    1-28     43-19#
CMDRST    1-29     39-6#
CMDSHT    1-33     45-6#
CMDSND    1-29     33-21#
CMDSPN    1-28     43-12#
CMDSPO    1-31     29-8#
CMDSQZ    1-32     26-9#
CMDTIM    1-30     38-5#
CMDYEL    1-28     33-6#
COAD      1-132    31-20
COAF      1-142    40-117
COAL      1-132    31-8
CONFIG    1-86     38-40
CORUSR    1-49     33-50     33-120    34-40     40-5      40-181    44-57
CPUAH     1-127    40-102*   40-146
CPUAL     1-127    40-103*   40-144
CR         2-5#    33-47     33-84
CRLF      1-125    11-71     12-53     22-50     28-47     36-28     40-88
CSABT    30-32#    30-34
CSCD     33-33     33-36#
CSEXIT   33-23     33-35#
```

Cross reference table (CREF V05.05)

```
CSGTOP  29-37    29-48#   29-51    32-91    32-97    32-103   32-109
CSHDEV   1-81    26-91
CSHDVN   1-81    26-116
CSIERR   8-67#
CSIMS1   1-124   28-55
CSIMS4   1-138   23-82
CSSCAN  33-29    33-43#
CTRLTT   1-77    33-14    34-35
CVTBUF  40-124   40-148   42-5#
CVTTAB   1-114    8-7     10-5     11-12    12-12    13-8     15-5     22-8     25-5     26-20    28-9     29-8
        32-114   35-5     37-5     38-5     43-23    44-6
CVTUC    1-113   12-13    15-6     29-9     32-115
CW$SOH   1-86    38-40
DAJMP   10-16#   10-30
DCCRD    1-103   39-9*
DCCWR    1-103   39-11*
DCTRD    1-103   39-8*
DCTWR    1-103   39-10*
DEADEV   1-120   25-77
DEAS1   10-7     10-20#
DELEMT   4-22#   30-20*   30-21
DELSPC   1-129   28-10
DETARG   1-142   35-28    35-30*   36-17
DETCHK   5-48    35-36#
DETGO   35-13    36-6#
DETHD    1-142    5-47#   35-16
DETIDL  35-38    35-41#   35-48
DETJMP  35-40    35-42#   35-50    36-29
DETKIL   5-49    35-46#
DEVIDL   1-131   30-33    32-12
DIVIDE   1-126   40-99
DJABMS   1-135   35-49
DLCEMT   1-28    12-44    12-64
DLMSG    1-135   36-26
DMTALL   1-135   40-76
DMTARG   1-119   23-26    25-59
DMTLGQ   5-38     5-40    25-87#
DMTNLG   4-16#   25-9*    25-47    25-86*
DMTNLQ   5-39     5-41    25-86#
DMTQHD   5-37#   25-16
DMTSUB   1-139   26-111
DMYDEV   1-48    29-34
DOASGN   1-66    11-87    23-106
DOMNT   23-111   24-8#
DOSTOP   1-136   40-201
EM$ALC   1-41    11-22    12-8
EM$ATF   1-59    11-66
EM$CSE   1-42    13-46
EM$DAA   1-62    11-59    12-51    22-48
EM$DET   1-53    35-8
EM$DIV   1-62    11-69
EM$DNR   1-41    26-33
EM$FNI   1-46    17-10
EM$FOE   1-85    36-41
EM$IAD   1-59    11-63    12-57    18-58
EM$ITF   1-46    19-19
```

Cross reference table (CREF V05.05)

```
EM$LDI    1-43     18-57
EM$NFW    1-41     26-16
EM$NID    1-60     30-24
EM$NLD    1-59     23-8
EM$NLN    1-43     34-19
EM$OLO    1-60     40-54
EM$SND    1-41     33-45
EM$SSY    1-60     26-44
EM$UAR    1-57     14-30
EM$UER    1-57     14-42
ERRLOC    1-52      8-67    11-56    12-48    22-45    24-13    28-40
FF        2-9#
FILNAM    1-116    16-22    18-34    18-38*   18-43    18-52    18-69    18-73    23-40    23-45    23-54    23-63
         36-11    44-15*   44-16*   44-17*   44-18*   44-19
FKILL     1-115     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    29-66    29-67    30-24    31-8
         33-16    33-45    34-19    34-37    35-8     35-19    35-25    36-33    36-37    36-41    37-50    38-50
         43-29    44-72
FORCEO    1-40     18-47    29-29
FPRINT    1-115    11-59    11-69    12-51    14-30    14-42    22-48    24-17    28-44    40-54
FSTDL     1-69     35-23
GAGMSG    1-134    33-107
GENMON    1-61     40-158
GTRD50    1-119    10-20    11-30    11-84    12-33    13-18    15-11    22-29    23-103   25-28    26-34    29-16
         37-24    44-14
IATTRB     4-12#    16-5*    16-35    21-31*
II$$SZ    1-146    18-78    19-13
II$FLG    1-147    16-35*
II$NAM    1-147    16-21    17-14*   18-66    18-71    18-73    19-11
II$NPV    1-147    16-39
II$PRV    1-147    16-37
IIBUF     1-47     16-21    16-35*   16-37    16-39    17-14*   18-66    18-71    18-73    19-11
ILLCMD    1-120    15-27    22-28    25-24
INDABT    1-115    40-39
INDERR    1-80     37-49*   38-49*
INDSAV    1-109    27-60
INDTSV    1-38     27-61
INFOMT    1-120    25-54
INQADR    1-47     18-63*   19-8*
INQEMT    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     17-15    20-7#
INSTBL    1-146    18-62    19-7
INSTBN    1-146    18-79    19-14
INVDAT    1-142    37-50
INVDEV    1-141    29-43
```

Cross reference table (CREF V05.05)

```
INVLDM    1-138    23-50
INVLDN    1-121    23-15
INVOPT    1-115    12-26    22-21    28-30    29-67    35-19
INVTIM    1-135    38-50
JOBCOM   43-6      43-13    43-23#
JS$OFF    1-61     40-159
KCSIBF    1-118     8-63     8-81     8-84*    8-87*    8-89    8-102    8-103    8-104
KCSIMS    1-119     8-69
KILEMT    1-136    40-183   43-5     44-59
KMNCHN    1-73     27-58
L         5-6       5-6#     5-7      5-7#     5-8      5-8#    5-9      5-9#     5-10     5-10#    5-11     5-11#
          5-12      5-12#    5-13     5-13#    5-14     5-14#   5-15     5-15#    5-16     5-16#    5-17     5-17#
          5-18      5-18#    5-19     5-19#    5-20     5-20#   5-21     5-21#    5-22     5-22#    5-23     5-23#
          5-30      5-30#    5-31     5-31#    5-38     5-38#   5-39     5-39#    5-40     5-40#    5-41     5-41#
          5-48      5-48#    5-49     5-49#    5-56     5-56#   5-57     5-57#    5-64     5-64#    6-7      6-7#
          6-8       6-8#     6-9      6-9#     6-10     6-10#   6-11     6-11#    6-12     6-12#    6-13     6-13#
          6-14      6-14#    6-15     6-15#    6-16     6-16#   6-17     6-17#    6-18     6-18#    6-19     6-19#
          6-20      6-20#    6-21     6-21#
LCONTM    1-71     40-138
LCPUHI    1-71     40-96
LCPULO    1-71     40-97
LD$RON    1-93     23-74
LDBASE    1-94     26-85
LDCLEN    1-58     23-116   25-81
LDDEVN   23-13     23-46    24-35#
LDDEVX    1-95     18-55    26-66    26-76
LDFLAG    1-94     23-75*
LDMNT     1-55     23-80
LDNAME    1-94     23-62    23-89    25-75*
LDPDEV    1-94     25-72*   26-84
LDSIZE    1-94     26-87
LF        2-4#      7-29    33-48    33-85    40-93    44-49
LF$WRT    1-112    40-72
LINFRE    1-135    35-41
LNPRIM    1-86     33-32    33-51    33-121   40-82
LNUMER   33-31     33-34#   33-37    33-91
LOFSPC    1-84     40-45    40-49    40-53*   40-61*
LOGASN    1-121    18-44    23-41
LOGCHK    1-101    23-21    25-43    26-123
LOGCLS    1-125    40-154
LOGFLG    1-111    40-72*
LPARNT    1-54     34-41    34-43
LPROG     1-70     40-121   40-203*
LPROJ     1-70     40-107   40-202*
LSECPT    1-74     41-25
LSTDL     1-69     35-26    40-84    40-173
LSTPL     1-102    33-119   40-15    40-171
LSTSL     1-104    33-30    40-180   43-27    44-56
LSTSPL    1-50     32-54
LSW       1-49     33-36    33-97    33-122   33-124   40-60*   40-80
LSW3      1-83     40-23    40-33*   40-41*   40-83*
LSW4      1-97      7-21    33-103   33-132   40-58*
LSW5      1-67      7-15    40-21
LSW6      1-108     7-17*    7-34*
LSW7      1-110    33-101   33-128
LSW9      1-66     40-25    40-59*   40-170*  40-198*
```

Cross reference table (CREF V05.05)

```
LUNAME    1-70      33-57
LWINDO    1-38      40-86
MAXSEC    1-74      41-26
MINTIM    1-74      40-137
MNBASE    1-129     28-12*
MNBPC     1-128     28-8*     28-22*
MNFLGS    1-128     28-7*     28-31*
MNTARG    1-138     23-110    26-61
MNTCSH   22-58       23-99     23-110#
MNTDEV    1-119     22-34*    22-41     23-20     23-105    25-38*    25-42     25-49     25-65     25-76     26-36*
MNTFUL    1-121     24-15
MNTLD    22-57      23-6#
MNTOP     1-129     28-16*
MONAR1    1-129     28-37
MONAR2    1-129     28-49
MONHD     1-129      5-55#    28-27
MSGBUF    1-134     33-46
MSGEND    1-134     33-81
MTOPHD    1-120      5-29#    22-18
MUL32     1-142     38-28     38-43     40-101
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      6-18      6-18      6-19      6-19      6-20      6-20      6-21      6-21
NFRESB    1-98      32-18
NMSGBF    1-148     33-114    33-140
NODEV    44-20      44-72#
NOFIL     1-133     32-50
NOFLAG    1-120     22-12*    22-22*    23-72
NOFRDL    1-142     36-33
NOOPTT    1-133     33-16     34-37
NOPMGN    1-129     28-42
NOTIM    40-108     40-118    40-154#
NOTON     1-134     33-99
NSPFLG    1-133     32-39
NUMON     1-65      44-31
NUMTB1    1-134     33-53
OFFEMT    1-136     40-204
OFFHD     5-63#     40-10
OFFNWF    4-17#     40-9*     40-17     41-5*
OFFNWN    5-64      41-5#
OPRCMD    1-31      33-14#
OPTLST    1-46      21-15     25-17     40-11
OTHRON    1-137     44-33
OTRMNT    1-139     26-56
PO$ALC    1-44      11-20     12-6
PO$BYP    1-44      26-14
PO$DET    1-53      35-6
PO$NFW    1-44      26-14
PO$SND    1-53      33-43
PAUMSG    1-114      7-27
PF$IOW    1-108      5-56
```

Cross reference table (CREF V05.05)

```
PF$SYS    1-108     5-57
PFCO      1-45     16-38
PFSO      1-45     16-36
PMBUSY    1-129    28-44
POPCF     1-124    40-52
PRGALL    1-27     40-38
PRIVCO    1-45     11-20     12-6      26-14     33-43     35-6      40-66
PRIVSO    1-44     40-67
PRTDEC    1-123    11-70     12-52     22-49     28-46     36-27     41-43
PRTTIM    1-141    40-92
PRTWRN    1-145    24-15     33-34     33-114    33-140
PRVOPT    1-47      5-18
PUSHCF    1-116    40-47
PVNPW     1-45     16-40     40-68
PVON      1-67     40-175
QHDMS1    1-60     32-52
QHDMS2    1-60     32-53
R.GSIZ    1-56     14-12*
R.GSTS    1-57     14-13*    14-17     14-23*    14-25     14-36*
R.NAME    1-56     14-14*    14-15*
R50ADD    4-9#     15-13
R50BUF    1-119    10-21     11-38     11-85     12-42     13-23     13-29     14-14     14-15     15-12     22-30     23-104
         25-29     26-35     29-20     29-23*    29-28     29-36     37-25     44-15
R50COM    1-84     36-7
R50DEL    4-10#    15-15
R50DSK    1-121    23-32
R50LD     1-121    22-31     24-35     25-35
R50LDO    1-119    22-33     24-37     24-38     24-42     25-37     25-66     25-70
R50LD7    1-121    24-40     25-68
R50MON    1-142    37-27
R50REM    4-11#    15-17
R50SAV    1-116    18-30
RDCMD     1-114     7-38      7-51      8-109     10-16     11-91     12-70     13-50     15-31     16-52     17-19     22-51
         23-120    25-82     28-48     28-51     29-71     30-25     30-32     31-22     31-74     31-86     31-97     32-55
         32-61     32-67     32-76     32-85     32-127    33-35     33-100    33-108    33-115    33-143    35-42     37-47
         38-47     39-19     40-29     40-62     43-45     44-51     45-9
REMRDB     4-8#    14-11
REMRGN    13-35    14-7#
RESDEV    1-107    40-34*
RJEMT     1-140    43-19
RS.CRR    1-56     14-17     14-25
RS.EGR    1-56     14-36
RS.GBL    1-56     14-13     14-23
RS.PVT    1-56     14-13
RSTVEC    39-13    39-23#
RUNMS     1-142    35-39
SD$BAK    1-97     31-96
SD$DEL    1-91     30-31
SD$FLG    1-92     32-37     32-90     32-96
SD$FLK    1-92     31-45     31-51     32-31
SD$SNG    1-98     32-14     32-60     32-66
SD$WFM    1-92     31-71     32-8
SD$WID    1-99     32-42     32-102    32-108
SDBU      1-97     31-94
SDCB      1-102    29-31
SDCBND    1-102    29-32
```

Cross reference table (CREF V05.05)

```
SDCBSZ    1-104     29-38
SDFHD     1-42      32-48
SDFLAG    1-92      30-31*    31-45*    31-51*    31-71*    31-96*    32-8      32-14     32-31     32-37     32-42     32-60*
          32-66*    32-90*    32-96*    32-102*   32-108*
SDFORM    1-92      31-53     32-24
SDNAME    1-48      29-34     29-36     31-17
SDSFCB    1-91      30-29     31-79     31-91     32-4
SDSKIP    1-97      31-85*
SEARCH    1-114     22-19     28-28     29-54     35-17
SEGCHN    1-38      27-56
SERFLG    1-49      23-25*    23-28*    23-90*    23-92*    25-58*    25-61*    40-48*    40-50*
SHTMSG    1-136     40-91
SJEMT     1-140     43-12
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
SNDALL    33-92     33-119#
SNDCOM    33-8      33-22#
SNDGN     33-15     33-30#
SNDLIN    33-27     34-10#
SNDONE    33-97#
SOPALC    1-50      11-16
SOPDAT    1-51      37-8
SOPTIM    1-51      38-8
SPACTV    1-131     32-6
SPCF      1-133     32-22
SPDLCR    30-10     30-29#
SPFLG     1-132     32-41
SPFLK     1-133     32-33
SPFUL     1-133     32-20
SPGEMT    1-133     32-136*   32-137
SPHLEM    4-28#     32-73*    32-74
SPLACT    1-136     40-191    44-36
SPLALN    6-7       31-5#
SPLBAK    6-8       31-90#
SPLBK1    31-92     31-94#
SPLCHN    1-72      27-57
SPLDEL    6-9       30-5#
SPLFLG    6-10      32-89#
SPLFRM    6-12      31-50#
SPLHD     1-130      6-6#     29-53
SPLHLD    6-13      32-72#
SPLLK     6-15      31-44#
SPLMT1    31-46     31-52#
SPLMUL    6-16      32-65#
SPLNFL    6-11      32-95#
SPLNHL    6-14      32-81#
SPLNRW    6-17      32-107#
SPLPND    1-137     44-38
SPLSKP    6-18      31-78#
SPLSNG    6-19      32-59#
SPLSTA    6-20      32-4#
SPLWID    6-21      32-101#
SPNARG    1-134     32-44
SPNBB     31-80     31-93#
SPNBSY    30-30     30-33#    31-93
```

Cross reference table (CREF V05.05)

```
SPNHEM    4-34#     32-82*    32-83
SPOLGO   31-73      32-135#
SPSNG     1-131     32-16
SPWFM     1-131     32-10
SPWIDE    1-134     32-46
SQZDEV    4-5#      26-37*    26-74     26-122
STPASK    1-137     44-43
STPFLG    1-72      40-19     40-89     40-164    44-55*    45-7*
SWPCHN    1-38      27-55
SYFLVC   27-24      27-55#
SYSDAT    1-105     37-46*
SYTIMH    1-105     38-45*
SYTIML    1-105     38-46*
TAB       2-8#
TALEMT    1-68      22-42
TIMSPL    4-40#     40-193
TIMUP    40-94      40-107#
TK1SEC    1-106     40-98
TM$SA1    1-37      41-40
TM$SA2    1-37      41-48
TM$SA3    1-37      41-42
TM$SA4    1-37      41-44
TMIDLH    1-58      39-29
TMIOH     1-58      39-26
TMIOWH    1-55      39-28
TMSWPH    1-58      39-27
TMSWTH    1-58      39-25
TMTOTH    1-55      39-23
TMUSRH    1-55      39-24
TOTON     1-72      40-199
TSKM2A    1-5#       1-26
UFORM     1-79      32-116    32-120    32-123
VLDSYS    1-145     23-6
WLDNAM    1-42       2-11#    18-24     18-69     18-71
XAREA     1-115     14-16     14-24     14-37     23-91     31-18     31-27     31-32     31-34     36-11     40-49     40-56
         40-115     40-122    40-149    40-193    44-19     44-21
YELEMT    1-135     33-7*     33-21*    33-105    33-109    33-130    33-134
```

...CM0	8-63	8-63	8-63	31-11	31-11	31-11	41-48	41-48	41-48	41-48		
...CM1	23-91	31-18	31-27	31-32	31-34	36-11	40-49	40-56	40-115	40-122	40-149	44-19
	44-21											
...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	36-11	36-11	40-49	40-49
	40-56	40-56	40-56	40-56	40-115	40-115	40-122	40-122	40-122	40-122	40-149	40-149
	40-149	40-149	40-193	44-19	44-19	44-21						
...CM3	31-21	31-38	31-39	36-13	40-150	44-25						
...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	32-36	32-39	32-41	32-44	32-46	32-50	32-52	32-53	33-99	33-107
	35-39	35-41	35-49	36-11	36-26	36-28	40-49	40-56	40-88	40-91	40-93	40-115
	40-117	40-122	40-149	40-193	41-40	41-42	41-44	44-19	44-21	44-33	44-38	44-43
...CM6	14-16	14-24	14-37	40-193								
...CM7	31-27	31-32	31-34	40-56	40-122	40-149						
.CLOSE	1-20#	31-21	31-38	31-39	36-13	40-150	44-25					
.CRRG	1-16#	14-16	14-24									
.CSIGE	1-18#	31-11										
.CSISP	1-16#	8-63										
.CSTAT	1-20#	44-21										
.ELRG	1-16#	14-37										
.ENTER	1-21#	31-18										
.FPROT	1-22#	23-91										
.GTLIN	1-19#	41-48										
.LOOKU	1-20#	36-11	40-49	40-115	44-19							
.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	32-39	32-41	32-44	32-46	32-50	32-52
	32-53	33-99	33-107	35-39	35-41	35-49	36-26	36-28	40-88	40-91	40-117	41-40
	41-42	41-44	44-33	44-38	44-43							
.READW	1-17#	31-32	40-56	40-122								
.TTYIN	1-17#	7-28	44-44									
.TTYOU	1-17#	32-27	32-36	40-93								
.TWAIT	1-17#	40-193										
.WRITW	1-21#	31-27	31-34	40-149								
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	6-18	6-19	6-20	6-21			
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	29-66	29-67	30-24	31-8
	33-16	33-45	34-19	34-37	35-8	35-19	35-25	36-33	36-37	36-41	37-50	38-50
	43-29	44-72										
FERR	3-4#	11-59	11-69	12-51	14-30	14-42	22-48	24-17	28-44	40-54		
FWARN	3-23#	24-15	33-34	33-114	33-140							
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	6-22				


```
TTTTT  SSS  K   K M   M  222  BBBB
  T   S   S K  K  MM MM 2   2 B   B
  T   S     K K   M M M     2 B   B
  T    SSS  KK    M   M   2   BBBB
  T       S K K   M   M  2    B   B
```