Table of contents

```
                        .TITLE  TSKM2C -- The RUN command
                        .ENABL  LC
                        .DSABL  GBL
000000                  .CSECT  TSKM2C
000000          TSKM2C:
                ;
                ;  TSKM2C is the portion of TSKMON that contains the code
                ;  to implement the RUN command.
                ;
                ;  Copyright 1986.
                ;  S&H Computer Systems, Inc.
                ;  Nashville, Tennessee
                ;
                ;
                ;  Macro calls
                ;
                        .MCALL  .LOOKUP,.READW,.CLOSE
                ;  Global definitions
                ;
                        .GLOBL  CMDRSY,CMDRUN,DORUN,RUNNAM,PLOAD,KDOCIN
                ;
                ;  Global references
                ;
                        .GLOBL  CMDOFF,NOPRG,FPRINT,SYNAME,INDSAV,NOIND
                        .GLOBL  $DEBUG,$HITTY,$INDRN,$NOINT,PO$DBG
                        .GLOBL  $NOWTT,$PRGLK,$RNIOP,$RNMLK,$TRNSP,$UCLRN
                        .GLOBL  AF$BYA,AF$CCA,AF$DBG,AF$DUP,AF$HIE,TRMSTR
                        .GLOBL  AF$IND,AF$IOP,AF$MEM,AF$NOI,AF$NOW,LSW4
                        .GLOBL  AF$PLK,AF$SCA,AF$SET,AF$TPO,AF$UCL,TRMSTR
                        .GLOBL  $CFOPN,CFBUF,CFSQEZ,PUSHCF,ACRFN,TOOLNG
                        .GLOBL  II$FLG,II$NPV,II$PRV,$QUIET
                        .GLOBL  DKSAV,SYSAV,CORUSR,EM$NPD,AF$NPWFILNAM
                        .GLOBL  VDBFLG,CINDAT,LOGASN,$STSNG,LSW6
                        .GLOBL  RUNARG,RUNCHN,RUNFLG,RUNDEV,INSSRC,PVNPW
                        .GLOBL  RUNHD,RUNEMT,JS$RUN,IIBUF
                        .GLOBL  LSW7,LITIME,PO$MEM,LSW2,LSW2S
                        .GLOBL  PRIVCO,EM$MPV,EM$NAL,PO$LOK
                        .GLOBL  CCSPRV,ABRTCF,PO$DBG,EM$NAD
                        .GLOBL  $NOWIN,LSW11,$DUPRN,IN$ACT,INDSTA
                        .GLOBL  IN$CNT,INDACT,UCLBLK,EM$NUC
                        .GLOBL  $SETRN,LSW9,RDERM,CINFLG,CVTTAB
                        .GLOBL  RSTPRV,LJSW,NEWJSW,R50VIR,VIMAGE,MAXMEM
                        .GLOBL  ODTBAS,VSWPFLMXJADR,PRGSIZ,PRGTOP
                        .GLOBL  USRSTK,NOSTRT,USTART,LPRG1,LPRG2,LMONHD
                        .GLOBL  SMONHD,GENMON,OVRCOR,BADSAV,LDNAM,LDCLEN
                        .GLOBL  NOCIN,SKPSPC,SEARCH,INVOPT,FKILL
                        .GLOBL  XAREA,FILNAM,LSW5,$SCCA,BLKO,BLKWDS
                        .GLOBL  LSW3,$CHACT,RDCMD,MXJADR,FIXPRV,AFCF
                ;
000000  000000  RUNOPS: .WORD   0                ;AF$xxx flags from RUN switches
```

```
                        ;-----------------------------------------------------
                        ;  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  CMDV2C
                        NAME'ID: .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  NAME2C
                        L     =       .
                              .ASCIZ  /STRNG/
                              .CSECT  CMDV2C
                              .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  CMDV2C
                                .WORD   0
                                .CSECT  TSKM2C
                                .ENDM   TBLEND
```

```
                                        .SBTTL  Tables of commands and options
                                ;-----------------------------------------------------------------
                                ;  Define option switches for the RUN command
                                ;
000002                                  TBLDEF  RUN,1
000002                                  CMDDEF  B*YPASN,RNBYA
000006                                  CMDDEF  D*EBUG,RNDBG
000012                                  CMDDEF  H*IGH,RNSHI
000016                                  CMDDEF  I*OPAGE,RNSIOP
000022                                  CMDDEF  L*OCK,RNSLK
000026                                  CMDDEF  M*EMLOCK,RNSMLK
000032                                  CMDDEF  NONI*NTERACTIVE,RNSNOI
000036                                  CMDDEF  NOW*INDOWING,RNNPW
000042                                  CMDDEF  O*DT,RNDBG
000046                                  CMDDEF  S*INGLECHAR,RNSCHR
000052                                  CMDDEF  SC*CA,RNSCCA
000056                                  CMDDEF  T*RANSPARENT,RNSTPO
000062                                  TBLEND
```

```
                                        .SBTTL  RUN command
                                ;-----------------------------------------------------------------
                                ;  Process the 'R' (RUN) command.
                                ;
                                ;  Entry point for 'RUN' command
                                ;
000002  012705  0000000         CMDRUN: MOV     #DKSAV,R5       ;SET DEFAULT DEV AND EXT
000006  000402                         BR      RUNPRS
                                ;
                                ;  Entry point for 'R' command
                                ;
000010  012705  0000000         CMDRSY: MOV     #SYSAV,R5       ;SET DEFAULT DEV AND EXT
                                ;
                                ;  Begin scanning command line
                                ;
000014  004767  0000000         RUNPRS: CALL    CVTTAB          ;CONVERT TABS TO SPACES IN COMMAND LINE
000020  005067  177754                  CLR     RUNOPS          ;Clear option flag word
                                ;
                                ;  See if any switches were specified with run command
                                ;
000024  004767  0000000         RUNSW:  CALL    SKPSPC          ;Skip over any spaces
000030  121327  000057                  CMPB    (R3),#'/        ;ANY SWITCH?
000034  001100                         BNE     RUNNAM          ;BR IF NOT
000036  005203                         INC     R3              ;POINT PAST SLASH
000040  012704  000000'                MOV     #RUNHD,R4       ;POINT TO RUN SWITCH TABLE
000044  004767  0000000                 CALL    SEARCH          ;ACCRUE AND LOOKUP SWITCH
000050  103004                         BCC     1$              ;BR IF SWITCH OK
000052                                  FABORT  #INVOPT         ;BAD SWITCH
000062  116701  0000000         1$:     MOVB    CORUSR,R1       ;PUT JOB INDEX NUMBER IN R1
000066  000134                         JMP     @(R4)+          ;ENTER SWITCH PROCESSING ROUTINE
                                ;
                                ;  Process the /LOCK switch
                                ;
000070  052767  0000000 177702  RNSLK:  BIS     #AF$PLK,RUNOPS  ;Remember to lock program to line
000076  000752                         BR      RUNSW           ;GO CHECK FOR OTHER SWITCHES
                                ;
                                ;  Process the /MEMLOCK switch
                                ;
000100  052767  0000000 177672  RNSMLK: BIS     #AF$MEM,RUNOPS  ;Remember to lock program in memory
000106  000746                         BR      RUNSW           ;Go check for more switches
                                ;
                                ;  Process the /SINGLECHAR switch.
                                ;
000110  052767  000000C 177662  RNSCHR: BIS     #<AF$SCA!AF$NOW>,RUNOPS ;Enable single char activ & non-wait
000116  000742                         BR      RUNSW
                                ;
                                ;  Process the /TRANSPARENT switch.
                                ;
000120  052767  0000000 177652  RNSTPO: BIS     #AF$TPO,RUNOPS  ;Say to run with transparent output
000126  000736                         BR      RUNSW
                                ;
                                ;  Process the /BYPASS (bypass assign) option
                                ;
000130  052767  0000000 177642  RNBYA:  BIS     #AF$BYA,RUNOPS  ;Bypass assign commands
000136  000732                         BR      RUNSW
                                ;
                                ;  Process the /HIGH (high-efficiency) switch.
```

```
                                ;
000140  052767  0000000 177632  RNSHI:  BIS     #AF$HIE,RUNOPS  ;Enable high-efficiency mode
000146  000726                         BR      RUNSW
                                ;
                                ;  Process the /IOPAGE switch
                                ;
000150  052767  0000000 177622  RNSIOP: BIS     #AF$IOP,RUNOPS  ;Program is to be run with I/O page access
000156  000720                         BR      RUNSW
                                ;
                                ;  Process the /NONINTERACTIVE switch
                                ;
000160  052767  0000000 177612  RNSNOI: BIS     #AF$NOI,RUNOPS  ;Run program in non-interactive mode
000166  000716                         BR      RUNSW
                                ;
                                ;  Process the /DEBUG switch
                                ;
000170  105767  0000000         RNDB:   TSTB    VDBFLG          ;Was program debugger genned into the system?
000174  001004                         BNE     1$              ;Br if yes
000176                                 FABORT  #EM$NPD         ;No program debugger available
000206  052767  0000000 177564  1$:     BIS     #AF$DBG,RUNOPS  ;Remember to run program with debugger
000214  000703                         BR      RUNSW
                                ;
                                ;  Process the /SCCA switch
                                ;
000216  052767  0000000 177554  RNSCCA: BIS     #AF$CCA,RUNOPS  ;Remember to suppress control-C aborts
000224  000677                         BR      RUNSW
                                ;
                                ;  Process the /NOWINDOW switch
                                ;
000226  052767  0000000 177544  RNNPW:  BIS     #AF$NPW,RUNOPS  ;Remember to suspend windowing
000234  000673                         BR      RUNSW
```

```
                                        ;
                                        ;  Accrue name of program to run
                                        ;
000236  004767  0000000         RUNNAM: CALL    ACRFN           ;ACCRUE THE PROGRAM FILE NAME
000242  103002                          BCC     1$              ;BR IF GOT NAME OK
000244  000167  001242                  JMP     CLSEXT          ;BR IF ERROR IN GETTING FILE NAME
000250  004767  002226          1$:     CALL    CHKIND          ;SEE IF WE ARE RUNNING IND
000254                                  .LOOKUP #XAREA,#RUNCHN,#FILNAM ;TRY TO OPEN FILE
000274  103020                          BCC     DORUNX          ;BRANCH IF OPEN OK
000276  032767  0000000 177474          BIT     #AF$PLK,RUNOPS  ;Was program supposed to be locked?
000304  001402                          BEQ     2$              ;NO, GO AHEAD
000306  000167  0000000                 JMP     CMDOFF          ;LOCKED FILE NOT FOUND, LOG OFF
000312                          2$:     FERR    #NOPRO          ;PRINT ERROR MESSAGE
000326  000167  001160                  JMP     CLSEXT          ;GO CLEANUP AND GET NEXT COMMAND
000332  005067  177442         DORUN:  CLR     RUNOPS          ;NO RUN OPTION FLAGS
000336  116701  0000000         DORUNX: MOVB    CORUSR,R1       ;GET LINE INDEX #
000342  016161  0000000 0000000         MOV     LSW2(R1),LSW2S(R1);SAVE VALUE OF LSW2
                                        ;
                                        ;  See if an argument was specified with run command
                                        ;
000350  012767  000001  0000100         MOV     #1,CINDAT+10    ;Set 1 to go in 510 chain area
000356  005067  0000120                 CLR     CINDAT+12       ;Set null at run argument string
000362  105267  0000000                 INCB    RUNARG          ;Set flag saying RUN argument string present
000366  112300                  1$:     MOVB    (R3)+,R0        ;ANY ARGUMENT FIELD PRESENT?
000370  001502                          BEQ     RNOARG          ;BR IF NOT
000372  120027  000040                  CMPB    R0,#' '         ;IGNORE SPACES
000376  001773                          BEQ     1$
                                        ;
                                        ;  There is an argument field.
                                        ;  Move argument string to chain area.
                                        ;
000400  005303                          DEC     R3              ;Point to start of string
000402  010302                          MOV     R3,R2           ;Get pointer to argument string
000404  012704  0000120                 MOV     #CINDAT+12,R4   ;Point to where chain data for 512 starts
000410  020427  0002700         7$:     CMP     R4,#CINDAT+270  ;Have we hit end of chain data area?
000414  103002                          BHIS    8$              ;Don't go past end of chain data area
000416  112224                          MOVB    (R2)+,(R4)+     ;Copy argument to chain data area
000420  001373                          BNE     7$              ;Loop till end reached
000422  162704  0000120         8$:     SUB     #CINDAT+12,R4   ;Get # chars in string (including null)
000426  010467  0000100                 MOV     R4,CINDAT+10    ;It will be passed in 510
                                        ;
                                        ;  Set argument up to be read like it was part of a command file.
                                        ;  If we have pending commands in the command file buffer, compress
                                        ;  them to make room for new commands.
                                        ;  If input is coming from a real command file, we do not need to
                                        ;  compress since we will reread buffer when we hit end of new command.
                                        ;
000432  032761  0000000 0000000         BIT     #$CFOPN,LSW4(R1);Is a command file open?
000440  001403                          BEQ     11$             ;Br if not
000442  012705  0010000                 MOV     #CFBUF+512.,R5  ;Say entire command file buffer is free
000446  000403                          BR      12$
000450  004767  0000000         11$:    CALL    CFSQEZ          ;Compress info in current command file buffer
000454  010005                          MOV     R0,R5           ;Save address of end of free space in buffer
                                        ;
                                        ;  Push current command file and start new one
                                        ;
000456  004767  0000000         12$:    CALL    PUSHCF          ;START NEW COMMAND FILE
```

```
      58 000462  012704  0000000                  MOV     #CFBUF,R4       ;POINT TO COMMAND FILE BUFFER
      59                                  ;
      60                                  ;  See if arg string contains a space which separates input file
      61                                  ;  specs from output file specs.
      62                                  ;
      63 000466  010302                   2$:     MOV     R3,R2           ;POINT TO START OF ARG STRING
      64 000470  112200                           MOVB    (R2)+,R0        ;GET CHAR FROM STRING
      65 000472  001416                           BEQ     3$              ;NO SPACE IN STRING
      66 000474  120027  000040                   CMPB    R0,#' '         ;SEARCH FOR SPACE
      67 000500  001373                           BNE     2$
      68                                  ;
      69                                  ;  String contains a space. Move part of string following space
      70                                  ;  to front of command file buffer and insert trailing '='.
      71                                  ;
      72 000502  032761  0000000 0000000          BIT     #$INDRN,LSW5(R1);Is IND being started?
      73 000510  001007                           BNE     3$              ;Br if yes -- Don't reverse parameters
      74 000512  105062  177777                   CLRB    -1(R2)          ;MAKE 1ST PART BE ASCIZ
      75 000516  112224                   4$:     MOVB    (R2)+,(R4)+     ;MOVE STRING TO COMMAND FILE BUFFER
      76 000520  001376                           BNE     4$              ;LOOP TILL DONE
      77 000522  112764  000075  177777           MOVB    #'=,-1(R4)      ;PUT IN SEPARATING =
      78                                  ;
      79                                  ;  Now move 1st part of string to command file buffer
      80                                  ;
      81 000530  112324                   3$:     MOVB    (R3)+,(R4)+
      82 000532  001376                           BNE     3$
      83                                  ;
      84                                  ;  Put in CR-LF-^C to terminate the line
      85                                  ;
      86 000534  012703  0000000                  MOV     #TRMSTR,R3      ;POINT TO STRING OF CONTROL CHARS
      87 000540  005304                           DEC     R4              ;STORE OVER TRAILING NULL
      88 000542  112324                   5$:     MOVB    (R3)+,(R4)+     ;PUT IN TERMINATING CHARS
      89 000544  001376                           BNE     5$
      90                                  ;
      91                                  ;  Pad rest of buffer with nulls
      92                                  ;
      93 000546  020405                   6$:     CMP     R4,R5           ;Reached end of area to fill?
      94 000550  001407                           BEQ     13$             ;Br if yes
      95 000552  103002                           BHIS    14$             ;Br if buffer overflow
      96 000554  105024                           CLRB    (R4)+           ;Null fill the buffer
      97 000556  000773                           BR      6$
      98 000560                           14$:    FABORT  #TOOLNG         ;Buffer overflow
      99                                  ;
     100                                  ;  Set flags to suppress listing of argument string being read.
     101                                  ;
     102 000570  052761  0000000 0000000  13$:    BIS     #$QUIET,LSW4(R1);SUPPRESS LISTING
     103                                  ;
     104                                  ;  Program file is open.
     105                                  ;
     106 000576  012704  0000000          RNOARG: MOV     #FILNAM,R4      ;POINT TO CELL WITH FILE NAME
     107 000602  016700  177172                   MOV     RUNOPS,R0       ;Get run option flags for PLOAD
     108 000606  000400                           BR      PLOAD           ;Load and start execution of program
```

```
                                ;-----------------------------------------------------------------------
                                ;  Load a SAV file into memory and start its execution.
                                ;
                                ;  Inputs:
                                ;   RUNCHN channel is now open to program file.
                                ;   R0 = AF$xxx run option flags.
                                ;   R1 = Job index number.
                                ;   R4 = Address of buffer with RAD50 program name.
                                ;
000610  010067  177164          PLOAD:  MOV     R0,RUNOPS       ;Save initial run option flags
                                ;
                                ;  Set name of running program
                                ;
000614  012705  0000000                 MOV     #RUNDEV,R5      ;Save program name here
000620  012700  000004                  MOV     #4,R0           ;Save 4 words
000624  012425                  1$:     MOV     (R4)+,(R5)+     ;Copy program file spec
000626  077002                          SOB     R0,1$
000630  012705  0000000                 MOV     #RUNDEV,R5      ;Point to file spec
000634  004767  0000000                 CALL    LOGASN          ;Do any logical assignment
000640  012704  0000000                 MOV     #RUNDEV,R4      ;Point to file spec
```

```
                                        ;  See if we need to invoke some automatic run switches for this program.
                                        ;
                                        ;  See if we should default to single character activation
                                        ;
000644  032761  0000000 0000000 RUNASW: BIT     #$STSNG,LSW6(R1);Is single char activation the default?
000652  001403                          BEQ     7$              ;Br if not
000654  052767  0000000 177116          BIS     #AF$SCA,RUNOPS  ;Remember to enable single char activation
                                        ;
                                        ;  See if the program has been installed
                                        ;
000662  012700  0000000         7$:     MOV     #RUNDEV,R0      ;Point to file spec for program
000666  004767  0000000                 CALL    INSSRC          ;See if this program has been installed
000672  103420                          BCS     6$              ;Br if not
                                        ;
                                        ;  Apply privilege flags specified for installed program
                                        ;
000674  012702  000000C                 MOV     #<PVNPW-1>*2,R2 ;Get index to last privilege word
000700  056262  000000C 0000000 4$:      BIS     II$PRV+IIBUF(R2),PRIVCO(R2);Set flags specified for program
000706  046262  000000C 0000000         BIC     II$NPV+IIBUF(R2),PRIVCO(R2);Clear flags specified for program
000714  162702  000002                  SUB     #2,R2           ;Get index to next word
000720  002367                          BGE     4$              ;Loop if more to do
000722  004767  0000000                 CALL    FIXPRV          ;Do privilege setup
000726  056767  000000C 177044          BIS     II$FLG+IIBUF,RUNOPS ;Combine run option switches for program
                                        ;
                                        ;  Or in attribute flags for enclosing command file
                                        ;
000734  056767  0000000 177036  6$:     BIS     AFCF,RUNOPS     ;Combine attributes from command file
                                        ;
                                        ;  Set some automatic switches for this program
                                        ;
000742  016702  177032                  MOV     RUNOPS,R2       ;Get attribute flags for program
000746  032702  0000000                 BIT     #AF$SCA,R2      ;Single character activation wanted?
000752  001403                          BEQ     3$              ;Br if not
000754  052761  0000000 0000000         BIS     #$CHACT,LSW5(R1);Enable single character activation
000762  032702  0000000         3$:     BIT     #AF$NOW,R2      ;Non wait TT input wanted?
000766  001403                          BEQ     15$             ;Br if not
000770  052761  0000000 0000000         BIS     #$NOWTT,LSW5(R1);Enable non-wait TT input
000776  032702  0000000         15$:    BIT     #AF$HIE,R2      ;Is high efficiency mode wanted?
001002  001403                          BEQ     16$             ;Br if not
001004  052761  0000000 0000000         BIS     #$HITTY,LSW4(R1);Enable high-efficiency mode
001012  032702  0000000         16$:    BIT     #AF$TPO,R2      ;Transparent output wanted?
001016  001403                          BEQ     23$             ;Br if not
001020  052761  0000000 0000000         BIS     #$TRNSP,LSW3(R1);Enable transparent output
001026  032702  0000000         23$:    BIT     #AF$NOI,R2      ;Is non-interactive execution wanted?
001032  001406                          BEQ     1$              ;Br if not
001034  052761  0000000 0000000         BIS     #$NOINT,LSW7(R1);Set non-interactive execution flag
001042  012761  000002  0000000         MOV     #2,LITIME(R1)   ;Set very short interactive time
001050  032702  0000000         1$:     BIT     #AF$IOP,R2      ;Map to I/O page?
001054  001417                          BEQ     17$             ;Br if not
001056  032767  0000000 0000000         BIT     #PO$MEM,PRIVCO  ;Are we authorized to access I/O page?
001064  001010                          BNE     20$             ;Br if yes
001066                                  FERR    #EM$MPV         ;Not authorized to run this program
001102  000167  000404                  JMP     CLSEXT
001106  052761  0000000 0000000 20$:     BIS     #$RNIOP,LSW9(R1);Set flag to cause mapping to I/O page
001114  032702  0000000         17$:    BIT     #AF$MEM,R2      ;Lock program in memory?
001120  001417                          BEQ     18$             ;Br if not
001122  032767  0000000 0000000         BIT     #PO$LOK,PRIVCO  ;Are we authorized to do this?
```

```
     58 001130  001010                          BNE     21$             ;Br if yes
     59 001132                                  FERR    #EM$NAL         ;Not authorized to lock program in memory
     60 001146  000167  000340                  JMP     CLSEXT
     61 001152  052761  0000000 0000000 21$:    BIS     #$RNMLK,LSW9(R1);Set flag to cause program to be locked
     62 001160  032702  0000000         18$:    BIT     #AF$PLK,R2      ;Want to lock program to line?
     63 001164  001407                          BEQ     22$             ;Br if not
     64 001166  052761  0000000 0000000         BIS     #$PRGLK,LSW5(R1);Remember program is locked to line
     65 001174  004767  0000000                 CALL    CCSPRV          ;Copy current to set privileges
     66 001200  004767  0000000                 CALL    ABRTCF          ;Close any open command files
     67 001204  032702  0000000         22$:    BIT     #AF$DBG,R2      ;Want to run program with debugger?
     68 001210  001417                          BEQ     24$             ;Br if not
     69 001212  032767  0000000 0000000         BIT     #PO$DBG,PRIVCO  ;Are we privileged to use the debugger?
     70 001220  001010                          BNE     19$             ;Br if yes
     71 001222                                  FERR    #EM$NAD         ;Can't use debugger
     72 001236  000167  000250                  JMP     CLSEXT
     73 001242  052761  0000000 0000000 19$:    BIS     #$DEBUG,LSW9(R1);Enable the debugger
     74 001250  032702  0000000         24$:    BIT     #AF$CCA,R2      ;Disable control-C abort?
     75 001254  001403                          BEQ     32$             ;Br if not
     76 001256  052761  0000000 0000000         BIS     #$SCCA,LSW5(R1) ;Disable control-C abort
     77 001264  032702  0000000         32$:    BIT     #AF$NPW,R2      ;Suspend process windowing?
     78 001270  001403                          BEQ     30$             ;Br if not
     79 001272  052761  0000000 0000000         BIS     #$NOWIN,LSW11(R1);Suspend process windowing
     80                                 ;
     81                                 ;  Check for startup of special programs
     82                                 ;
     83 001300  032702  0000000         30$:    BIT     #AF$DUP,R2      ;Is DUP being started?
     84 001304  001403                          BEQ     25$             ;Br if not
     85 001306  052761  0000000 0000000         BIS     #$DUPRN,LSW6(R1);Remember DUP is running
     86 001314  032702  0000000         25$:    BIT     #AF$IND,R2      ;Is IND being started?
     87 001320  001423                          BEQ     26$             ;Br if not
     88 001322  132767  0000000 0000000         BITB    #IN$ACT,INDSTA  ;Is IND already active?
     89 001330  001414                          BEQ     27$             ;Br if not
     90 001332  132767  0000000 0000000         BITB    #IN$CNT,INDSTA  ;Are we continuing its execution?
     91 001340  001010                          BNE     27$             ;Br if yes
     92 001342                                  FERR    #INDACT         ;Error -- IND is already active
     93 001356  000167  000130                  JMP     CLSEXT
     94 001362  052761  0000000 0000000 27$:    BIS     #$INDRN,LSW5(R1);Remember IND is running
     95 001370  032702  0000000         26$:    BIT     #AF$UCL,R2      ;Is TSXUCL being started?
     96 001374  001412                          BEQ     28$             ;Br if not
     97 001376  005767  0000000                 TST     UCLBLK          ;Are we supporting user-defined commands?
     98 001402  001004                          BNE     31$             ;Br if yes
     99 001404                                  FABORT  #EM$NUC         ;Not supporting user-defined commands
    100 001414  052761  0000000 0000000 31$:    BIS     #$UCLRN,LSW7(R1);Remember TSXUCL is running
    101 001422  032702  0000000         28$:    BIT     #AF$SET,R2      ;Is SETUP being started?
    102 001426  001403                          BEQ     2$              ;Br if not
    103 001430  052761  0000000 0000000         BIS     #$SETRN,LSW9(R1);Remember SETUP is running
    104 001436                          2$:
```

```
                                        ;
                                        ;  Read in block zero of file being started.
                                        ;
001436                          RNREAD: .READW  #XAREA,#RUNCHN,#BLK0,#BLKWDS,#0
001474  103045                          BCC     SVINBG          ;BRANCH IF READ OK
001476                                  FERR    #RDERM
001512                          CLSEXT: .CLOSE  #RUNCHN         ;CLOSE SAV FILE
001520  105067  0000000                 CLRB    CINFLG          ;RESET CHAIN FLAG
001524  105067  0000000                 CLRB    RUNARG          ;No argument string with RUN command
001530  116701  0000000                 MOVB    CORUSR,R1       ;GET CURRENT JOB INDEX NUMBER
001534  042761  0000000 0000000         BIC     #$TRNSP,LSW3(R1)
001542  042761  0000000 0000000         BIC     #$HITTY,LSW4(R1);CLEAN OUT MORE RUN FLAGS
001550  042761  0000000 0000000         BIC     #<$INDRN!$CHACT!$NOWTT>,LSW5(R1);CLEAN OUT FLAGS
001556  042761  0000000 0000000         BIC     #$DUPRN,LSW6(R1)
001564  042761  0000000 0000000         BIC     #$UCLRN!$NOINT,LSW7(R1)
001572  042761  0000000 0000000         BIC     #$DEBUG!$RNIOP!$RNMLK!$SETRN,LSW9(R1)
001600  004767  0000000                 CALL    RSTPRV          ;Reset job privilege flags
001604  000167  0000000                 JMP     RDCMD
                                        ;
                                        ;  Examine information in block 0 of sav file.
                                        ;
001610  116701  0000000         SVINBG: MOVB    CORUSR,R1       ;GET JOB INDEX NUMBER
001614  012702  0000000                 MOV     #BLK0,R2        ;POINT TO BLOCK 0 DATA WE JUST READ IN
                                        ;  Set up new JSW for job being started.
001620  016203  000044                  MOV     44(R2),R3       ;GET JSW SPECIFIED IN SAV FILE IMAGE
001624  042703  100007                  BIC     #^C<077770>,R3  ;CLEAR SOME FLAGS
001630  016104  0000000                 MOV     LJSW(R1),R4     ;GET CURRENT JSW
001634  042704  075570                  BIC     #075570,R4      ;CLEAR SOME FLAGS IN IT
001640  050304                          BIS     R3,R4           ;MERGE FLAGS
001642  010467  0000000                 MOV     R4,NEWJSW       ;THIS WILL BE JSW FOR JOB WE ARE STARTING
001646  026267  000000  0000000         CMP     0(R2),R50VIR    ;WAS JOB LINKED WITH /V SWITCH?
001654  001410                          BEQ     23$             ;IF YES THEN SET VIRTUAL-MODE FLAG
001656  026227  000056  000034          CMP     56(R2),#28.     ;DOES JOB NEED MORE THAN 28KW?
001664  101004                          BHI     23$             ;BR IF YES
001666  026227  000050  160000          CMP     50(R2),#160000  ;CHECK ADDRESS ABOVE TOP OF PROGRAM ROOT
001674  101403                          BLOS    27$             ;BR IF DOES NOT NEED MORE THAN 28KW
001676  052767  0000000 0000000 23$:    BIS     #VIMAGE,NEWJSW  ;FORCE VIRTUAL-JOB MODE IF > 28KW NEEDED
                                        ;  Set up information about the max amount of memory job will be allowed
                                        ;  to use.  This is constrained by three things:
                                        ;    1. The amount of memory specified by the MEMORY command.
                                        ;    2. 56Kb if this is not a virtual job, (64Kb if it is a virtual job).
001704  016705  0000000         27$:    MOV     MAXMEM,R5       ;Get amt of memory specified by MEMORY command
001710  032767  0000000 0000000         BIT     #VIMAGE,NEWJSW  ;Is this a virtual job?
001716  001005                          BNE     17$             ;Br if this is a virtual job
001720  020527  160000                  CMP     R5,#160000      ;Constrain to 56Kb if not virtual job
001724  101402                          BLOS    17$
001726  012705  160000                  MOV     #160000,R5
001732  010567  0000000         17$:    MOV     R5,ODTBAS       ;This is base address of debugger
                                        ;  If stack is allocated above top of program then say top of program
                                        ;  is equal to top of stack.
001736  026262  000042  000050          CMP     42(R2),50(R2)   ;IS STACK ALLOCATED ABOVE PROGRAM TOP?
001744  101403                          BLOS    16$             ;BR IF NOT
001746  016262  000042  000050          MOV     42(R2),50(R2)   ;SAY TOP OF PROGRAM = STACK TOP
                                        ;  See if amount of memory to allocate for program was specified
                                        ;  in SAV file.
001754  016703  0000000         16$:    MOV     MAXMEM,R3       ;GET DEFAULT TOP OF MEMORY ADDRESS
001760  105767  0000000                 TSTB    VSWPFL          ;IS THIS A NON-SWAPPING SYSTEM?
```

```
     58 001764  001442                          BEQ     13$             ;BR IF YES
     59 001766  032767  0000000 0000000         BIT     #VIMAGE,NEWJSW  ;IS THIS A VIRTUAL IMAGE?
     60 001774  001005                          BNE     26$             ;BR IF YES
     61 001776  020327  160000                  CMP     R3,#160000      ;CONSTRAIN TO 56KB IF NOT VIRTUAL
     62 002002  101402                          BLOS    26$
     63 002004  012703  160000                  MOV     #160000,R3
     64 002010  016200  000056          26$:    MOV     56(R2),R0       ;WAS MEMORY ALLOCATION FOR PROGRAM SPECIFIED?
     65 002014  001406                          BEQ     7$              ;BR IF NOT
     66 002016  072027  000013                  ASH     #11.,R0         ;CONVERT # KW TO BYTE ADDRESS
     67 002022  001002                          BNE     21$             ;BR IF NOT 64KB
     68 002024  012700  177774                  MOV     #177774,R0      ;SET 64KB ADDRESS TOP
     69 002030  010003                  21$:    MOV     R0,R3           ;SET AS ADDRESS OF TOP OF PROGRAM SPACE
     70 002032  016200  000050          7$:     MOV     50(R2),R0       ;GET ADDRESS OF TOP OF STATIC PROGRAM AREA
     71 002036  062700  000002                  ADD     #2,R0
     72 002042  001002                          BNE     22$             ;BR IF DIDN'T OVERFLOW 64KB
     73 002044  012700  177774                  MOV     #177774,R0      ;SET SIZE FOR 64KB
     74 002050  020300                  22$:    CMP     R3,R0           ;IS ALLOCATED SIZE AT LEAST THIS BIG?
     75 002052  103001                          BHIS    8$              ;BR IF YES
     76 002054  010003                          MOV     R0,R3           ;USE ENOUGH SPACE FOR STATIC PROGRAM SIZE
     77 002056  016705  0000000         8$:     MOV     MXJADR,R5       ;GET MAX ADDRESS JOB CAN POSSIBLY USE
     78 002062  010367  0000000                 MOV     R3,ODTBAS       ;START ODT ABOVE TOP OF PROGRAM AREA
     79 002066  020305                          CMP     R3,R5           ;ROOM FOR PROGRAM TO RUN?
     80 002070  101123                          BHI     TOOBIG          ;BR IF NOT ENOUGH MEMORY
     81 002072  032767  0000000 0000000 13$:    BIT     #VIMAGE,NEWJSW  ;IS THIS A VIRTUAL JOB?
     82 002100  001005                          BNE     25$             ;BR IF YES
     83 002102  020327  160000                  CMP     R3,#160000      ;IS NOT, THEN CONSTRAIN SIZE TO 56KB
     84 002106  101402                          BLOS    25$
     85 002110  012703  160000                  MOV     #160000,R3
     86 002114  010367  0000000         25$:    MOV     R3,PRGSIZ       ;ADDRESS ABOVE TOP OF PROGRAM AREA
     87                                 ;  Check top of program address.
     88 002120  016203  000050                  MOV     50(R2),R3       ;GET TOP-OF-PROGRAM ADDRESS FOR SAV FILE
     89 002124  020327  001000                  CMP     R3,#1000        ;MAKE SURE IT'S NOT TOO SMALL
     90 002130  103512                          BLO     ILLSAV          ;BR IF INVALID
     91 002132  010367  0000000                 MOV     R3,PRGTOP       ;SAVE TOP ADDRESS OF PROGRAM
     92 002136  016700  0000000                 MOV     ODTBAS,R0       ;GET TOP ADDRESS WE MAY GO UP TO
     93 002142  020300                          CMP     R3,R0           ;IS PROGRAM TOO BIG?
     94 002144  103075                          BHIS    TOOBIG          ;BR IF YES
     95                                 ;  Check program's stack address.
     96 002146  016203  000042                  MOV     42(R2),R3       ;GET SPECIFIED STARTING STACK ADDRESS
     97 002152  001407                          BEQ     2$              ;BR IF NO INITIAL STACK ADDRESS SPECIFIED
     98 002154  020367  0000000                 CMP     R3,PRGTOP       ;CAN'T BE ABOVE TOP OF PROGRAM
     99 002160  101076                          BHI     ILLSAV
    100 002162  032703  000001                  BIT     #1,R3           ;CAN'T BE ODD
    101 002166  001073                          BNE     ILLSAV
    102 002170  000402                          BR      3$
    103 002172  012703  001000         2$:     MOV     #1000,R3        ;DEFAULT TO 1000 FOR STACK IF NONE SPECIFIED
    104 002176  010367  0000000         3$:     MOV     R3,USRSTK       ;SET AS STARTUP STACK ADDRESS
    105                                ;  Check program starting address.
    106 002202  016203  000040                  MOV     40(R2),R3       ;GET PROGRAM START ADDRESS
    107 002206  020327  000400                  CMP     R3,#400         ;MAKE SURE IT'S VALID
    108 002212  103406                          BLO     4$              ;BR IF TOO LOW
    109 002214  032703  000001                  BIT     #1,R3           ;MAKE SURE IT'S NOT ODD
    110 002220  001003                          BNE     4$
    111 002222  020367  0000000                 CMP     R3,PRGTOP       ;MAKE SURE IT IS IN MEMORY RANGE FOR PROGRAM
    112 002226  101410                          BLOS    5$
    113 002230                         4$:     FERR    #NOSTRT         ;INVALID START ADDRESS
    114 002244  000167  177242                  JMP     CLSEXT
```

```
                                        5$:     MOV     R3,USTART       ;SAVE PROGRAM STARTING ADDRESS
                                        ;
                                        ;  Set up name of program that is being started
                                        ;
                                                MOV     #RUNDEV,R4      ;POINT TO PROGRAM NAME STORAGE CELLS
                                                MOV     2(R4),LPRG1(R1) ;SAVE NAME OF RUNNING PROGRAM
                                                MOV     4(R4),LPRG2(R1)
                                        ;
                                        ;  Set up status flags for program being started
                                        ;
                                                MOV     RUNOPS,RUNFL0   ;Set status flags for running program
                                        ;
                                        ;  Broadcast status message to any monitoring jobs telling them that
                                        ;  we are starting execution of a job.
                                        ;
                                                TST     LMONHD(R1)      ;Is our job being monitored?
                                                BNE     29$             ;Br if it is
                                                TST     SMONHD          ;Anyone monitoring all jobs?
                                                BEQ     28$             ;Br if not
                                        29$:    MOV     #GENMON,R0      ;Point to EMT argument block
                                                MOV     #JS$RUN,2(R0)   ;Set status code
                                                EMT     375             ;Broadcast status message
                                        ;
                                        ;  Enter TSX to read in SAV file and transfer control to it.
                                        ;
                                        28$:    MOV     #RUNEMT,R0      ;EMT TO START PROGRAM EXECUTION
                                                EMT     375             ;START SAV FILE
                                        ;
                                        ;  PROGRAM IS TOO BIG TO FIT IN MEMORY.
                                        ;
                                        TOOBIG: FERR    #OVRCOR
                                                BR      CLEJMP
                                        ;
                                        ;  BAD SAV FILE FORMAT
                                        ;
                                        ILLSAV: FERR    #BADSAV
                                        CLEJMP: JMP     CLSEXT
```

```
                                ;-----------------------------------------------
                                ;  PROCESS A .CHAIN REQUEST
                                ;
002376  012704  000000G       KDOCIN: MOV     #CINDAT,R4      ;POINT TO CELL WITH PROGRAM NAME
                                ;
                                ;  If chain is being done to SY:LD.SYS, simply do a SET LD CLEAN
                                ;
002402  010405                          MOV     R4,R5           ;POINT TO PROGRAM NAME
002404  012703  000000G                 MOV     #LDNAM,R3       ;POINT TO "SY:LD.SYS"
002410  012700  000004                  MOV     #4,R0           ;COMPARE 4 WORDS
002414  022325                  3$:     CMP     (R3)+,(R5)+     ;COMPARE PROGRAM NAMES
002416  001004                          BNE     2$              ;BR IF NOT CHAINING TO LD HANDLER
002420  077003                          SOB     R0,3$
002422  004767  000000G                 CALL    LDCLEN          ;DO "SET LD CLEAN" OPERATION
002426  000421                          BR      4$              ;DON'T DO THE CHAIN
                                ;
                                ;  Look up SAV file we are chaining to
                                ;
002430                          2$:     .LOOKUP #XAREA,#RUNCHN,R4
002446  103403                          BCS     1$              ;BRANCH IF ERROR
002450  005000                          CLR     R0              ;Say no RUN options
002452  000167  176132                  JMP     PLOAD           ;LOAD AND RUN PROGRAM
002456                          1$:     FERR    #NOPRG
002472  105067  000000G         4$:     CLRB    CINFLG
002476  000167  000000G                 JMP     NOCIN
                                ;
```

```
                                        .SBTTL  CHKIND -- Force IND to run from boot device
                                ;-------------------------------------------------------------
                                ;
                                ;  Inputs:  FILNAM has dev:filspc.ext of program about to be run
                                ;
                                ;  Outputs:  If program is IND, dev: in FILNAM changed to boot device
                                ;                               and .ext in FILNAM changed to .SAV
                                ;
002502  026727  0000020 035164  CHKIND:  CMP     FILNAM+2,#^RIND ;IS PROGRAM IND?
002510  001021                          BNE     9$              ;RETURN IF NOT
002512  005767  0000040                 TST     FILNAM+4        ;AND ONLY 3 CHARS?
002516  001016                          BNE     9$              ;RETURN IF NOT
                                ;  Program about to be run is IND
002520  005767  0000000                 TST     INDSAV          ;WAS IND AVAILABLE DURING START-UP?
002524  001005                          BNE     1$              ;BR IF SO
002526  005726                          TST     (SP)+           ;THROW AWAY RETURN POINTER IF NOT
002530                                  FABORT  #NOIND          ;AND ABORT COMMAND
                                ;  Ensure that IND is always run from "real" SY
002540  016767  0000000 0000000 1$:     MOV     SYNAME,FILNAM   ;ALWAYS RUN FROM BOOT DEVICE
002546  012767  073376  0000060         MOV     #^RSAV,FILNAM+6 ;USE IND.SAV
002554  000207                  9$:     RETURN
                                ;
000001                                  .END
Errors detected:  0

*** Assembler statistics

Work  file  reads:  0
Work  file writes:  0
Size of work file:  9848 Words  ( 39 Pages)
Size of core pool:  17920 Words  ( 70 Pages)
Operating  system:  RT-11

Elapsed time:  00:00:16.22
DK:TSKM2C,LP:TSKM2C=DK:TSKM2C.MAC/C/N:SYM
```

Cross reference table (CREF V05.04)

```
$CFOPN    1-30     5-48
$CHACT    1-48     7-34     8-13
$DEBUG    1-25     7-73     8-16
$DUPRN    1-39     7-85     8-14
$HITTY    1-25     7-40     8-12
$INDRN    1-25     5-72     7-94     8-13
$NOINT    1-25     7-46     8-15
$NOWIN    1-39     7-79
$NOWTT    1-26     7-37     8-13
$PRGLK    1-26     7-64
$QUIET    1-31     5-102
$RNIOP    1-26     7-54     8-16
$RNMLK    1-26     7-61     8-16
$SCCA     1-47     7-76
$SETRN    1-41     7-103    8-16
$STSNG    1-33     7-5
$TRNSP    1-26     7-43     8-11
$UCLRN    1-26     7-100    8-15
...V1     5-8      5-8      5-8      8-4      8-4      8-7      8-7      9-19     9-19     9-19
...V2     5-8      5-8      5-8#     5-8#     8-4      8-4      8-4#     8-4#     8-7      8-7#     9-19     9-19
          9-19#    9-19#
ABRTCF    1-38     7-66
ACRFN     1-30     5-4
AF$BYA    1-27     4-54
AF$CCA    1-27     4-82     7-74
AF$DBG    1-27     4-77     7-67
AF$DUP    1-27     7-83
AF$HIE    1-27     4-59     7-38
AF$IND    1-28     7-86
AF$IOP    1-28     4-64     7-48
AF$MEM    1-28     4-39     7-55
AF$NOI    1-28     4-69     7-44
AF$NOW    1-28     4-44     7-35
AF$NPW    1-32     4-87     7-77
AF$PLK    1-29     4-34     5-10     7-62
AF$SCA    1-29     4-44     7-7      7-32
AF$SET    1-29     7-101
AF$TPO    1-29     4-49     7-41
AF$UCL    1-29     7-95
AFCF      1-48     7-27
BADSAV    1-45     8-150
BLKO      1-47     8-4      8-23
BLKWDS    1-47     8-4
CCSPRV    1-38     7-65
CFBUF     1-30     5-50     5-58
CFSQEZ    1-30     5-52
CHKIND    5-7      10-9#
CINDAT    1-33     5-21*    5-22*    5-34     5-35     5-39     5-40*    9-4
CINFLG    1-41     8-8*     9-24*
CLEJMP    8-146    8-151#
CLSEXT    5-6      5-14     7-53     7-60     7-72     7-93     8-7#     8-114    8-151
CMDOFF    1-24     5-12
CMDRSY    1-20     4-12#
CMDRUN    1-20     4-7#
CORUSR    1-32     4-29     5-16     8-10     8-22
CVTTAB    1-41     4-16
```

Cross reference table (CREF V05.04)

```
DKSAV    1-32    4-7
DORUN    1-20    5-15#
DORUNX   5-9     5-16#
EM$MPV   1-37    7-52
EM$NAD   1-38    7-71
EM$NAL   1-37    7-59
EM$NPD   1-32    4-76
EM$NUC   1-40    7-99
FILNAM   1-47    5-8     5-106   10-9    10-11   10-19*  10-20*
FIXPRV   1-48    7-22
FKILL    1-46    4-28    4-76    5-98    7-99    10-17
FPRINT   1-24    5-13    7-52    7-59    7-71    7-92    8-6     8-113   8-145   8-150   9-23
GENMON   1-45    8-134
II$FLG   1-31    7-23
II$NPV   1-31    7-19
II$PRV   1-31    7-18
IIBUF    1-35    7-18    7-19    7-23
ILLSAV   8-90    8-99    8-101   8-150#
IN$ACT   1-39    7-88
IN$CNT   1-40    7-90
INDACT   1-40    7-92
INDSAV   1-24   10-14
INDSTA   1-39    7-88    7-90
INSSRC   1-34    7-12
INVOPT   1-46    4-28
JS$RUN   1-35    8-135
KDOCIN   1-20    9-4#
L        3-6     3-6#    3-7     3-7#    3-8     3-8#    3-9     3-9#    3-10    3-10#   3-11    3-11#
         3-12    3-12#   3-13    3-13#   3-14    3-14#   3-15    3-15#   3-16    3-16#   3-17    3-17#
LDCLEN   1-45    9-14
LDNAM    1-45    9-9
LITIME   1-36    7-47*
LJSW     1-42    8-27
LMONHD   1-44    8-130
LOGASN   1-33    6-19
LPRG1    1-44    8-120*
LPRG2    1-44    8-121*
LSW11    1-39    7-79*
LSW2     1-36    5-17
LSW2S    1-36    5-17*
LSW3     1-48    7-43*   8-11*
LSW4     1-28    5-48    5-102*  7-40*   8-12*
LSW5     1-47    5-72    7-34*   7-37*   7-64*   7-76*   7-94*   8-13*
LSW6     1-33    7-5     7-85*   8-14*
LSW7     1-36    7-46*   7-100*  8-15*
LSW9     1-41    7-54*   7-61*   7-73*   7-103*  8-16*
MAXMEM   1-42    8-42    8-56
MXJADR   1-48    8-77
NARGS    3-5#    3-6     3-6     3-7     3-7     3-8     3-8     3-9     3-9     3-10    3-10    3-11
         3-11    3-12    3-12    3-13    3-13    3-14    3-14    3-15    3-15    3-16    3-16    3-17
         3-17
NEWJSW   1-42    8-30*   8-37*   8-43    8-59    8-81
NOCIN    1-46    9-25
NOIND    1-24   10-17
NOPRG    1-24    5-13    9-23
NOSTRT   1-44    8-113
```

Cross reference table (CREF V05.04)

```
ODTBAS   1-43    8-48*   8-78*   8-92
OVRCOR   1-45    8-145
PO$DBG   1-25    1-38    7-69
PO$LOK   1-37    7-57
PO$MEM   1-36    7-50
PLOAD    1-20    5-108   6-10#   9-22
PRGSIZ   1-43    8-86*
PRGTOP   1-43    8-91*   8-98    8-111
PRIVCO   1-37    7-18*   7-19*   7-50    7-57    7-69
PUSHCF   1-30    5-57
PVNPW    1-34    7-17
R50VIR   1-42    8-31
RDCMD    1-48    8-18
RDERM    1-41    8-6
RNBYA    3-6     4-54#
RNDB     3-7     3-14    4-74#
RNNPW    3-13    4-87#
RNOARG   5-25    5-106#
RNREAD   8-4#
RNSCCA   3-16    4-82#
RNSCHR   3-15    4-44#
RNSHI    3-8     4-59#
RNSIOP   3-9     4-64#
RNSLK    3-10    4-34#
RNSMLK   3-11    4-39#
RNSNOI   3-12    4-69#
RNSTPO   3-17    4-49#
RSTPRV   1-42    8-17
RUNARG   1-34    5-23*   8-9*
RUNASW   7-5#
RUNCHN   1-34    5-8     5-8     8-4     8-4     8-7     8-7     9-19    9-19
RUNDEV   1-34    6-14    6-18    6-20    7-11    8-119
RUNEMT   1-35    8-140
RUNFLG   1-34    8-125*
RUNHD    1-35    3-5#    4-25
RUNNAM   1-20    4-23    5-4#
RUNOPS   1-50#   4-17*   4-34*   4-39*   4-44*   4-49*   4-54*   4-59*   4-64*   4-69*   4-77*   4-82*
         4-87*   5-10    5-15*   5-107   6-10*   7-7*    7-23*   7-27*   7-31    8-125
RUNPRS   4-8     4-16#
RUNSW    4-21#   4-35    4-40    4-45    4-50    4-55    4-60    4-65    4-70    4-78    4-83    4-88
SEARCH   1-46    4-26
SKPSPC   1-46    4-21
SMONHD   1-45    8-132
SVINBG   8-5     8-22#
SYNAME   1-24    10-19
SYSAV    1-32    4-12
TOOBIG   8-80    8-94    8-145#
TOOLNG   1-30    5-28
TRMSTR   1-27    1-29    5-86
TSKM2C   1-5#
UCLBLK   1-40    7-97
USRSTK   1-44    8-104*
USTART   1-44    8-115*
VDBFLG   1-33    4-74
VIMAGE   1-42    8-37    8-43    8-59    8-81
VSWPFL   1-43    8-57
XAREA    1-47    5-8     8-4     9-19
```

Cross reference table (CREF V05.04)

...CM1	5-8	8-4	9-19									
...CM2	5-8	5-8	8-4	8-4	8-4	8-4	9-19	9-19				
...CM3	8-7											
...CM5	5-8	8-4	9-19									
...CM7	8-4											
.CLOSE	1-17#	8-7										
.LOOKU	1-17#	5-8	9-19									
.READW	1-17#	8-4										
CMDDEF	2-47#	3-6	3-7	3-8	3-9	3-10	3-11	3-12	3-13	3-14	**3-15**	**3-16**
	3-17											
FABORT	2-15#	4-28	4-76	5-98	7-99	10-17						
FERR	2-4#	5-13	7-52	7-59	7-71	7-92	8-6	8-113	8-145	8-150	**9-23**	
FWARN	2-23#											
TBLDEF	2-35#	3-5										
TBLEND	2-61#	3-18										