```
   SSSSSSSSSSSS  DDDDDDDDDDDD        AAAAAAAAA
   SSSSSSSSSSSS  DDDDDDDDDDDD        AAAAAAAAA
   SSSSSSSSSSSS  DDDDDDDDDDDD        AAAAAAAAA
SSS              DDD         DDD  AAA         AAA
SSS              DDD         DDD  AAA         AAA
SSS              DDD         DDD  AAA         AAA
SSS              DDD         DDD  AAA         AAA
SSS              DDD         DDD  AAA         AAA
SSS              DDD         DDD  AAA         AAA
   SSSSSSSSS     DDD         DDD  AAA         AAA
   SSSSSSSSS     DDD         DDD  AAA         AAA
   SSSSSSSSS     DDD         DDD  AAA         AAA
            SSS  DDD         DDD  AAAAAAAAAAAAAAA
            SSS  DDD         DDD  AAAAAAAAAAAAAAA
            SSS  DDD         DDD  AAAAAAAAAAAAAAA
            SSS  DDD         DDD  AAA         AAA
            SSS  DDD         DDD  AAA         AAA
            SSS  DDD         DDD  AAA         AAA
SSSSSSSSSSSS     DDDDDDDDDDDD     AAA         AAA
SSSSSSSSSSSS     DDDDDDDDDDDD     AAA         AAA
SSSSSSSSSSSS     DDDDDDDDDDDD     AAA         AAA
```

```
_$

Sy
--
CT
CT
CT
CT
CU
CU
CU
DC
DC
DC
DC
DC
DC
DE
DE
DE
DE
DE
DE
DE
DE
DI
DI
DI
DI
DI
DI
DI
DI
DI
DI
DI
DI
DI
DU
DU
DU
DU
DU
DU
DU
DU
DV
DV
DV
DY
DY
DY
DY
DY
ER
ES
```

```
                                                M 15
**FILE**ID**PARSE









          PPPPPPPP      AAAAAA    RRRRRRRR      SSSSSSSS  EEEEEEEEEE
          PPPPPPPP      AAAAAA    RRRRRRRR      SSSSSSSS  EEEEEEEEEE
          PP      PP  AA      AA  RR      RR  SS          EE
          PP      PP  AA      AA  RR      RR  SS          EE
          PP      PP  AA      AA  RR      RR  SS          EE
          PP      PP  AA      AA  RR      RR  SS          EE
          PPPPPPPP    AA      AA  RRRRRRRR      SSSSSS    EEEEEEEE
          PPPPPPPP    AA      AA  RRRRRRRR      SSSSSS    EEEEEEEE
          PP          AAAAAAAAAA  RR  RR              SS  EE
          PP          AAAAAAAAAA  RR  RR              SS  EE
          PP          AA      AA  RR    RR            SS  EE              ....
          PP          AA      AA  RR    RR            SS  EE              ....
          PP          AA      AA  RR      RR  SSSSSSSS    EEEEEEEEEE      ....
          PP          AA      AA  RR      RR  SSSSSSSS    EEEEEEEEEE      ....


          LL            IIIIII      SSSSSSSS
          LL            IIIIII      SSSSSSSS
          LL              II      SS
          LL              II      SS
          LL              II      SS
          LL              II      SS
          LL              II        SSSSSS
          LL              II        SSSSSS
          LL              II              SS
          LL              II              SS
          LL              II              SS
          LL              II              SS
          LLLLLLLLLL    IIIIII    SSSSSSSS
          LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
0000          .TITLE  PARSE           Parse tables for SDA commands
0000          .SBTTL  COPYRIGHT NOTICE
0000          .IDENT  'V04-000'
0000    ;
0000    ;****************************************************************************
0000    ;*                                                                          *
0000    ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000    ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000    ;*  ALL RIGHTS RESERVED.                                                    *
0000    ;*                                                                          *
0000    ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000    ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000    ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000    ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000    ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000    ;*  TRANSFERRED.                                                            *
0000    ;*                                                                          *
0000    ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000    ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000    ;*  CORPORATION.                                                            *
0000    ;*                                                                          *
0000    ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000    ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000    ;*                                                                          *
0000    ;*                                                                          *
0000    ;****************************************************************************
0000    ;
```

```
0000            .SBTTL  PROGRAM DESCRIPTION
0000    ;++
0000    ; FACILITY
0000    ;
0000    ;       SYSTEM DUMP ANALYZER
0000    ;
0000    ; ABSTRACT
0000    ;
0000    ;       This module contains the parse tables which
0000    ;       are used to parse SDA commands.
0000    ;
0000    ; ENVIRONMENT
0000    ;
0000    ;       NATIVE MODE, USER MODE
0000    ;
0000    ; AUTHOR
0000    ;
0000    ;       TIM HALVORSEN, JULY 1978
0000    ;
0000    ; MODIFIED BY
0000    ;
0000    ;       V03-015 EMB0109         Ellen M. Batbouta       24-Jul-1984
0000    ;               Fix SHOW DEVICE/ADDRESS so that it will be consistent
0000    ;               with other sda commands with the qualifier /address
0000    ;               such as SHOW PORTS/ADDRESS.
0000    ;
0000    ;       V03-014 EMB0102         Ellen M. Batbouta       30-May-1984
0000    ;               Add support for SHOW CLUSTER, SHOW CONNECTIONS, SHOW
0000    ;               RSPID, and SHOW PORTS.
0000    ;
0000    ;       V03-013 PRB0302         Paul Beck       28-Dec-1983  17:50
0000    ;               Add support for EXAMINE/INST/NOSKIP
0000    ;
0000    ;       V03-012 ROW0260         Ralph O. Weber          21-NOV-1983
0000    ;               Changes parse tables to allow SHOW RESOURCES, SHOW LOCKS, and
0000    ;               SHOW PROCESS /LOCKS.
0000    ;
0000    ;       V03-011 ROW0237         Ralph O. Weber          10-NOV-1983
0000    ;               Add TPARSE table entries to support the SHOW DEVICE/ADDR
0000    ;               command.
0000    ;
0000    ;       V03-010 WMC0002         Wayne Cardoza           30-Sep-1983
0000    ;               Bad transition after page table range
0000    ;
0000    ;       V03-009 BLS0230         Benn Schreiber          24-Jul-1983
0000    ;               Correct end of line / comment handling
0000    ;
0000    ;       V03-008 WMC0001         Wayne cardoza           11-Jul-1983
0000    ;               Support for separate display of P0 and P1 process page tables.
0000    ;
0000    ;       V03-007 JLV0259         Jake VanNoy             23-MAY-1983
0000    ;               ADD DEFINE/KEY.
0000    ;
0000    ;       V03-006 TMK0001         Todd M. Katz            21-Mar-1983
0000    ;               ADD SET LOG and SET NOLOG syntax.
0000    ;
0000    ;       V03-005 CWH1002         CW Hobbs                 2-Mar-1983
```

```
                                       0000     86 ;               Store only the low bits of proc index (as determined by
                                       0000     87 ;               the global PIX_WIDTH so that we can interchange internal
                                       0000     88 ;               and extended pids.  Change syntax of SHOW SUMMARY to add
                                       0000     89 ;               SHOW SUMMARY /IMAGE.
                                       0000     90 ;
                                       0000     91 ;       V03-004 RAS0123         Ron Schaefer                8-Feb-1983
                                       0000     92 ;               Add SHOW PROC/RMS=RJB syntax.
                                       0000     93 ;
                                       0000     94 ;       V03-003 JLV0224         Jake VanNoy                21-JAN-1983
                                       0000     95 ;               Add Examine/PSL Examine/CONDITION_VLAUE and Examine/TIME
                                       0000     96 ;               parsing. Add VALIDATE QUEUE syntax.
                                       0000     97 ;
                                       0000     98 ;       V03-002 LMP0031         L. Mark Pilant,            17-Jun-1982  13:15
                                       0000     99 ;               Add support for SHOW PROCESS/CHANNELS.  Also, change the parsing
                                       0000    100 ;               to allow a process name along with qualifiers.
                                       0000    101 ;
                                       0000    102 ;       V03-001 MSH0001         Maryann Hinden             10-Jun-1982
                                       0000    103 ;               Add SHOW LOCK, SHOW RES, SHOW PROC/LOCK commands.
                                       0000    104 ;
                                       0000    105 ;--
```

```
                0000   107          .SBTTL  DECLARATIONS
                0000   108 ;
                0000   109 ;        SYMBOL DEFINTIONS
                0000   110 ;
                0000   111          $TPADEF                         ; TPARSE DEFINITIONS
                0000   112          $OPTDEF                         ; DEFINE BITS IN OPTIONS WORD
                0000   113          $SMGDEF                         ; SCREEN INPUT DEFINITIONS
                0000   114
       0000003B 0000   115 SEMI     = ^A';'
       0000003A 0000   116 COLON    = ^A':'
```

```
                0000            .SBTTL  DATA STORAGE
                0000    ;
                0000    ;       STORAGE DEFINITIONS
                0000    ;
                0000
            00000000            .PSECT  SDADATA,WRT,NOEXE
                0000
                0000    PREV_BUFFER:
       00000050 0000            .BLKB   80                      ; HOLDS PREVIOUS COMMAND
                0050    PREV_COMMAND:
       00000000 0050            .LONG   0
       00000000' 0054           .ADDRESS PREV_BUFFER
                0058
                0058    KEY_ATTR:
       00000002 0058            .LONG   SMG$M_KEY_TERMINATE     ; KEYPAD ATTRIBUTE FOR "REPEAT KEY"
```

```
005C              .SBTTL  PARSER STATE TABLE
005C       ;
005C       ;      STATE TABLE FOR DUMP ANALYZER COMMANDS
005C       ;
005C
005C              $INIT_STATE SDA_STATE,SDA_KEY
005C
005C              $STATE  START
005C              $TRAN   'READ',READ_CMD
005C              $TRAN   'REPEAT',START,REPEAT_COMMAND
005C              $TRAN   TPA$_LAMBDA,,SAVE_COMMAND
005C              $STATE
005C              $TRAN   '@',INDIRECT_CMD
005C              $TRAN   'COPY',COPY_CMD
005C              $TRAN   'DEFINE',DEFINE
005C              $TRAN   'EXAMINE',EXAMINE
005C              $TRAN   'FORMAT',FORMAT_CMD
005C              $TRAN   'HELP',HELP_CMD
005C              $TRAN   'SET',SET
005C              $TRAN   'SEARCH',SEARCH
005C              $TRAN   'SHOW',SHOW
005C              $TRAN   'EVALUATE',EVALUATE
005C              $TRAN   'EXIT',EXIT_CMD
005C              $TRAN   'VALIDATE',VALIDATE
005C              $TRAN   TPA$_LAMBDA,CK_EOS
005C
005C              $STATE  CK_EOS
005C              $TRAN   '!',TPA$_EXIT
005C              $TRAN   TPA$_EOS,TPA$_EXIT
005C
005C       ;
005C       ;      SEARCH start:end = object
005C       ;             start;length = object
005C       ;
005C
005C              $STATE  SEARCH
005C              $TRAN   !EXPRESSION
005C              $STATE
005C              $TRAN   SEMI,,,OPT$M_LENGTH,OPTIONS
005C              $TRAN   COLON,,,OPT$M_RANGE,OPTIONS
005C              $STATE
005C              $TRAN   !EXPRESSION
005C              $STATE
005C              $TRAN   '='
005C              $TRAN   TPA$_LAMBDA
005C              $STATE
005C              $TRAN   !EXPRESSION
005C              $STATE
005C              $TRAN   TPA$_LAMBDA,TPA$_EXIT,SEARCH_MEMORY
005C
005C       ;
005C       ;      @filespec
005C       ;
005C
005C              $STATE  INDIRECT_CMD
005C              $TRAN   !COLLECT,TPA$_EXIT,INDIRECT_COMMAND,,FILE_DESC
005C
```

```
005C  ;
005C  ;          SHOW PROCESS name /options
005C  ;          SHOW SUMMARY
005C  ;          SHOW PAGE_TABLE
005C  ;          SHOW PFN_DATA
005C  ;          SHOW POOL /IRP /NONPAGED /ALL
005C  ;          SHOW CRASH
005C  ;          SHOW STACK /mode   (default is current stack)
005C  ;          SHOW SYMBOL name /ALL
005C  ;          SHOW DEVICE device /ALL
005C  ;          SHOW RMS
005C  ;          SHOW LOCKS
005C  ;          SHOW RESOURCES
005C  ;          SHOW CLUSTER /CSID=n /SCS
005C  ;          SHOW CONNECTIONS /ADDRESS=n
005C  ;          SHOW PORTS /ADDRESS=n
005C  ;          SHOW RSPID /CONNECTION=n
005C  ;
005C
005C             $STATE  SHOW
005C             $TRAN   'CRASH',TPA$_EXIT,DISPLAY_CRASH
005C             $TRAN   'DEVICE',DEVICE
005C             $TRAN   'HEADER',TPA$_EXIT,PRINT_HEADER
005C             $TRAN   'LOCKS',LOCK
005C             $TRAN   'PAGE_TABLE',PAGE_TABLE
005C             $TRAN   'PFN_DATA',PFN_DATA
005C             $TRAN   'POOL',POOL
005C             $TRAN   'PROCESS',PROCESS
005C             $TRAN   'RESOURCES',RES
005C             $TRAN   'RMS',SHOW_RMS
005C             $TRAN   'STACKS',STACK
005C             $TRAN   'SUMMARY',SHOW_SUMMARY
005C             $TRAN   'SYMBOLS',SYMBOL
005C             $TRAN   'CLUSTER',VAXCLUSTER
005C             $TRAN   'CONNECTIONS',SCS_CONNECT
005C             $TRAN   'RSPID',SCS_RDT
005C             $TRAN   'PORTS',SCS_PORTS
005C  ;
005C  ;          SHOW LOCK lockid
005C  ;                    [/ALL]
005C  ;
005C
005C             $STATE  LOCK
005C             $TRAN   !LOCK_OPTS,TPA$_EXIT
005C             $TRAN   TPA$_HEX,,,,LOCKID
005C
005C             $STATE
005C             $TRAN   !CK_EOS,TPA$_EXIT,SHOW_ONE_LOCK
005C
005C             $STATE  LOCK_OPTS
005C             $TRAN   !CK_EOS,TPA$_EXIT,SHOW_ALL_LOCKS
005C             $TRAN   '/'
005C
005C             $STATE
005C             $TRAN   'ALL'
005C
005C             $STATE
```

```
005C        $TRAN   !CK_EOS,TPA$_EXIT,SHOW_ALL_LOCKS
005C
005C  ;
005C  ;     SHOW RES [/ALL]
005C  ;              /LOCKID = xxxx
005C  ;
005C
005C        $STATE  RES
005C        $TRAN   !CK_EOS,TPA$_EXIT,SHOW_ALL_RES
005C        $TRAN   '/'
005C
005C        $STATE
005C        $TRAN   'ALL',TPA$_EXIT,SHOW_ALL_RES
005C        $TRAN   'LOCKID'
005C
005C        $STATE
005C        $TRAN   ':'
005C        $TRAN   '='
005C
005C        $STATE
005C        $TRAN   TPA$_HEX,,,,LOCKID
005C
005C        $STATE
005C        $TRAN   !CK_EOS,TPA$_EXIT,SHOW_ONE_RES
005C
005C  ;
005C  ;     SHOW RMS
005C  ;
005C
005C        $STATE  SHOW_RMS
005C         $TRAN  !CK_EOS,TPA$_EXIT,SHOW_RMS_OPT
005C
005C  ;     SHOW CLUSTER /CSID=n /SCS
005C
005C        $STATE  VAXCLUSTER
005C        $TRAN   !CK_EOS,TPA$_EXIT,SHOW_CLUSTER
005C        $TRAN   '/'
005C
005C        $STATE
005C        $TRAN   'CSID'
005C        $TRAN   'SCS',TPA$_EXIT,SHOW_SCS
005C
005C        $STATE
005C        $TRAN   '='
005C
005C        $STATE
005C        $TRAN   TPA$_HEX,,,,CSID
005C
005C        $STATE
005C        $TRAN   !CK_EOS,TPA$_EXIT,SHOW_CLUSTER
005C
005C  ;
005C  ;     SHOW CONNECTIONS /ADDR=n
005C  ;
005C
005C        $STATE  SCS_CONNECT
005C        $TRAN   !CK_EOS,TPA$_EXIT,SHOW_CONNECTIONS
```

```
        005C                    $TRAN   '/'
        005C
        005C                    $STATE
        005C                    $TRAN   'ADDRESS'
        005C
        005C                    $STATE
        005C                    $TRAN   '='
        005C
        005C                    $STATE
        005C                    $TRAN   !EXPRESSION,TPA$_EXIT,CDT_BYADDR
        005C            ;
        005C            ;       SHOW PORTS /ADDR=n
        005C            ;
        005C
        005C                    $STATE  SCS_PORTS
        005C                    $TRAN   !CK_EOS,TPA$_EXIT,SHOW_PORTS
        005C                    $TRAN   '/'
        005C
        005C                    $STATE
        005C                    $TRAN   'ADDRESS'
        005C
        005C                    $STATE
        005C                    $TRAN   '='
        005C
        005C                    $STATE
        005C                    $TRAN   !EXPRESSION,TPA$_EXIT,PDT_BYADDR
        005C            ;
        005C            ;       SHOW RSPID /CONNECTION=n
        005C            ;
        005C
        005C                    $STATE  SCS_RDT
        005C                    $TRAN   !CK_EOS,TPA$_EXIT,SHOW_RSPID
        005C                    $TRAN   '/'
        005C
        005C                    $STATE
        005C                    $TRAN   'CONNECTION'
        005C
        005C                    $STATE
        005C                    $TRAN   '='
        005C
        005C                    $STATE
        005C                    $TRAN   !EXPRESSION,,,1,CDT_SPCFY
        005C
        005C                    $STATE
        005C                    $TRAN   TPA$_LAMBDA,TPA$_EXIT,SHOW_RSPID
        005C
        005C
        005C            ;
        005C            ;       SET LOG filespec
        005C            ;       SET NOLOG
        005C            ;       SET OUTPUT filespec
        005C            ;       SET PROCESS name /INDEX=n
        005C            ;       SET RMS/DISPLAY=(opt1,opt2,...,optn)
        005C            ;
        005C
        005C                    $STATE  SET
        005C                    $TRAN   'LOG',SET_LOG
        005C    362             $TRAN   'NOLOG',TPA$_EXIT,CLOSE_LOG
        005C    363             $TRAN   'OUTPUT',SET_OUTPUT
        005C    364             $TRAN   'PROCESS',SET_PROC
        005C    365             $TRAN   'RMS',SET_RMS
        005C    366
        005C    367     ;
        005C    368     ;       SET LOG filespec
        005C    369     ;
        005C    370
        005C    371             $STATE  SET_LOG
        005C    372             $TRAN   !COLLECT,,,,LOG_FILE
        005C    373             $STATE
        005C    374             $TRAN   TPA$_LAMBDA,TPA$_EXIT,OPEN_LOG
        005C    375
        005C    376     ;
        005C    377     ;       SET OUTPUT filespec
        005C    378     ;
        005C    379
        005C    380             $STATE  SET_OUTPUT
        005C    381             $TRAN   !COLLECT,,,,OUTPUT_FILE
        005C    382             $STATE
        005C    383             $TRAN   TPA$_LAMBDA,TPA$_EXIT,OPEN_OUTPUT
        005C    384
        005C    385     ;
        005C    386     ;       SUB-EXPRESSION TO COLLECT ALL REMAINING CHARACTERS
        005C    387     ;
        005C    388
        005C    389             $STATE  COLLECT
        005C    390             $TRAN   TPA$_ANY                        ; MUST HAVE AT LEAST 1 CHAR.
        005C    391             $STATE  COLLECT_LOOP
        005C    392             $TRAN   !SNARF,COLLECT_LOOP
        005C    393             $TRAN   TPA$_LAMBDA,TPA$_EXIT
        005C    394             $STATE  SNARF
        005C    395             $TRAN   !CK_EOS,TPA$_FAIL
        005C    396             $TRAN   TPA$_ANY,TPA$_EXIT
        005C    397
        005C    398     ;
        005C    399     ;       SET RMS=opt
        005C    400     ;               (opt1,opt2,...,optn)
        005C    401     ;
        005C    402
        005C    403             $STATE  SET_RMS
        005C    404              $TRAN  !DISPLAY,TPA$_EXIT,END_SET_RMS
        005C    405
        005C    406             $STATE  DISPLAY
        005C    407              $TRAN  TPA$_LAMBDA,,SETUP_RMS_TMP
        005C    408             $STATE
        005C    409              $TRAN  ':'
        005C    410              $TRAN  '='
        005C    411             $STATE
        005C    412              $TRAN  !DIS_OPT,TPA$_EXIT
        005C    413              $TRAN  '('
        005C    414             $STATE  NXT_OPT
        005C    415              $TRAN  !DIS_OPT
        005C    416             $STATE
        005C    417              $TRAN  <','>,NXT_OPT
        005C    418              $TRAN  ')',TPA$_EXIT
```

```
005C    419
005C    420             $STATE  DIS_OPT
005C    421              $TRAN  TPA$_LAMBDA,,SETUP_DIS_OPT
005C    422             $STATE
005C    423              $TRAN  !ATM_OPT,DIS_OPT_EXIT
005C    424              $TRAN  'N'
005C    425             $STATE
005C    426              $TRAN  'O'
005C    427             $STATE
005C    428              $TRAN  !ATM_OPT,,,OPT$M_NO,RMS_DIS_TMP1
005C    429             $STATE  DIS_OPT_EXIT
005C    430              $TRAN  TPA$_LAMBDA,TPA$_EXIT,STORE_TMP1_OPT
005C    431
005C    432             $STATE  ATM_OPT
005C    433              $TRAN  'ALL',IFI,,OPT$M_RMSALL,RMS_DIS_TMP1
005C    434              $TRAN  'ASB',TPA$_EXIT,,OPT$M_ASB,RMS_DIS_TMP1
005C    435              $TRAN  'BDB',TPA$_EXIT,,OPT$M_BDB,RMS_DIS_TMP1
005C    436              $TRAN  'BDBSUM',TPA$_EXIT,,OPT$M_BDBSUM,RMS_DIS_TMP1
005C    437              $TRAN  'BLB',TPA$_EXIT,,OPT$M_BLB,RMS_DIS_TMP1
005C    438              $TRAN  'BLBSUM',TPA$_EXIT,,OPT$M_BLBSUM,RMS_DIS_TMP1
005C    439              $TRAN  'CCB',TPA$_EXIT,,OPT$M_CCB,RMS_DIS_TMP1
005C    440              $TRAN  'FAB',TPA$_EXIT,,OPT$M_FAB,RMS_DIS_TMP1
005C    441              $TRAN  'FCB',TPA$_EXIT,,OPT$M_FCB,RMS_DIS_TMP1
005C    442              $TRAN  'FWA',TPA$_EXIT,,OPT$M_FWA,RMS_DIS_TMP1
005C    443              $TRAN  'GBD',TPA$_EXIT,,OPT$M_GBD,RMS_DIS_TMP1
005C    444              $TRAN  'GBDSUM',TPA$_EXIT,,OPT$M_GBDSUM,RMS_DIS_TMP1
005C    445              $TRAN  'GBH',TPA$_EXIT,,OPT$M_GBH,RMS_DIS_TMP1
005C    446              $TRAN  'IDX',TPA$_EXIT,,OPT$M_IDX,RMS_DIS_TMP1
005C    447              $TRAN  'IFAB',IFI,,OPT$M_IFB,RMS_DIS_TMP1
005C    448              $TRAN  'IFB',IFI,,OPT$M_IFB,RMS_DIS_TMP1
005C    449              $TRAN  'IRAB',TPA$_EXIT,,OPT$M_IRB,RMS_DIS_TMP1
005C    450              $TRAN  'IRB',TPA$_EXIT,,OPT$M_IRB,RMS_DIS_TMP1
005C    451              $TRAN  'NAM',TPA$_EXIT,,OPT$M_NAM,RMS_DIS_TMP1
005C    452              $TRAN  'RAB',TPA$_EXIT,,OPT$M_RAB,RMS_DIS_TMP1
005C    453              $TRAN  'RJB',TPA$_EXIT,,OPT$M_RJB,RMS_DIS_TMP1
005C    454              $TRAN  'RLB',TPA$_EXIT,,OPT$M_RLB,RMS_DIS_TMP1
005C    455              $TRAN  'TRC',TPA$_EXIT,,OPT$M_TRC,RMS_DIS_TMP1
005C    456              $TRAN  'WCB',TPA$_EXIT,,OPT$M_WCB,RMS_DIS_TMP1
005C    457              $TRAN  'XAB',TPA$_EXIT,,OPT$M_XAB,RMS_DIS_TMP1
005C    458              $TRAN  '*',TPA$_EXIT,STORE_STAR
005C    459
005C    460             $STATE  IFI
005C    461              $TRAN  ':'
005C    462              $TRAN  '='
005C    463              $TRAN  TPA$_LAMBDA,TPA$_EXIT
005C    464             $STATE  IFI2
005C    465              $TRAN  'ALL',TPA$_EXIT,CLR_IFI_TMP
005C    466              $TRAN  !EXPRESSION,TPA$_EXIT,STORE_IFI_TMP
005C    467
005C    468     ;
005C    469     ;       EVALUATE expression
005C    470     ;
005C    471
005C    472             $STATE  EVALUATE
005C    473             $TRAN    '/'
005C    474             $TRAN   TPA$_LAMBDA,EVAL_EXP
005C    475             $STATE
```

```
005C             $TRAN    'CONDITION_VALUE',,,OPT$M_COND,OPTIONS
005C             $STATE   EVAL_EXP
005C             $TRAN    !EXPRESSION
005C             $STATE
005C             $TRAN    '/',EVAL_QUAL
005C             $TRAN    TPA$_LAMBDA,TPA$_EXIT,SHOW_EXPR
005C             $STATE   EVAL_QUAL
005C             $TRAN    'CONDITION_VALUE',,,OPT$M_COND,OPTIONS
005C             $STATE
005C             $TRAN    TPA$_LAMBDA,TPA$_EXIT,SHOW_EXPR
005C
005C     ;
005C     ;       EXAMINE start:end /P0 /P1 /SYSTEM /ALL /INSTRUCTION /PSL /TIME /CONDITION
005C     ;       EXAMINE start;length
005C     ;
005C
005C             $STATE   EXAMINE
005C             $TRAN    '/'
005C             $TRAN    TPA$_LAMBDA,EXAMINE_EXPRESSION
005C             $STATE
005C             $TRAN    'P0',EXAM_OK,,OPT$M_P0,OPTIONS
005C             $TRAN    'P1',EXAM_OK,,OPT$M_P1,OPTIONS
005C             $TRAN    'SYSTEM',EXAM_OK,,OPT$M_SYSTEM,OPTIONS
005C             $TRAN    'ALL',EXAM_OK,,OPT$M_ALL,OPTIONS
005C             $TRAN    'INSTRUCTION',EXAM_INS,,OPT$M_INST,OPTIONS
005C             $TRAN    'PSL',,,OPT$M_PSL,OPTIONS
005C             $TRAN    'TIME',,,OPT$M_TIME,OPTIONS
005C             $TRAN    'CONDITION_VALUE',,,OPT$M_COND,OPTIONS
005C
005C             $STATE   EXAMINE_EXPRESSION
005C             $TRAN    !CK_EOS,EXAM_OK
005C             $TRAN    !EXPRESSION
005C             $STATE
005C             $TRAN    !CK_EOS,EXAM_OK
005C             $TRAN    '/',EXAM_QUALS
005C             $TRAN    SEMI,,,OPT$M_LENGTH,OPTIONS
005C             $TRAN    COLON,,,OPT$M_RANGE,OPTIONS
005C             $STATE
005C             $TRAN    !EXPRESSION
005C             $STATE
005C             $TRAN    '/',EXAM_QUALS
005C             $TRAN    TPA$_LAMBDA,EXAM_OK
005C             $STATE   EXAM_QUAL_LOOP
005C             $TRAN    !CK_EOS,EXAM_OK
005C             $TRAN    '/'
005C             $STATE   EXAM_QUALS
005C             $TRAN    'INSTRUCTION',EXAM_QUAL_LOOP,,OPT$M_INST,OPTIONS
005C             $TRAN    'NOSKIP',EXAM_QUAL_LOOP,,OPT$M_NOSKIP,OPTIONS
005C             $TRAN    'PSL',EXAM_QUAL_LOOP,,OPT$M_PSL,OPTIONS
005C             $TRAN    'TIME',EXAM_QUAL_LOOP,,OPT$M_TIME,OPTIONS
005C             $TRAN    'CONDITION_VALUE',EXAM_QUAL_LOOP,,OPT$M_COND,OPTIONS
005C             $STATE   EXAM_OK
005C             $TRAN    TPA$_LAMBDA,TPA$_EXIT,EXAM_MEMORY
005C             $STATE   EXAM_INS
005C             $TRAN    '/'
005C             $TRAN    TPA$_LAMBDA,EXAMINE_EXPRESSION
005C             $STATE
```

```
005C           $TRAN   'NOSKIP',EXAM_INS,,OPT$M_NOSKIP,OPTIONS
005C    ;
005C    ; VALIDATE QUEUE expression
005C    ;
005C           $STATE  VALIDATE
005C           $TRAN   'QUEUE',VAL_QUEUE
005C
005C           $STATE  VAL_QUEUE
005C           $TRAN   !CK_EOS,VALIDATE_Q_OK
005C           $TRAN   !VAL_QUEUE_OPTS,TPA$_EXIT
005C
005C           $STATE  VAL_QUEUE_OPTS
005C           $TRAN   '/',VAL_QUE_QUAL
005C           $TRAN   !EXPRESSION,VAL_QUEUE
005C
005C           $STATE  VAL_QUE_QUAL
005C           $TRAN   'SELF_RELATIVE',VAL_QUEUE,,OPT$M_SELF,OPTIONS
005C           $TRAN   'MAXIMUM_LINKS'
005C           $STATE
005C           $TRAN   ':'
005C           $TRAN   '='
005C           $STATE
005C           $TRAN   !DECIMAL,VAL_QUEUE,VAL_SET_MAX
005C
005C           $STATE  VALIDATE_Q_OK
005C           $TRAN   TPA$_LAMBDA,TPA$_EXIT,VALIDATE_QUEUE
005C
005C    ;
005C    ;      EXPRESSION ANALYSIS FOR EXAMINE ARGUMENTS
005C    ;
005C    ;      THIS REDUCTION TABLE USES A SEPARATE STACK FOR PARSING
005C    ;      TOKENS AS THEY ENCOUNTERED.  THE RESULT OF THE LAST
005C    ;      OPERATION IS ALWAYS PLACED IN TPA$L_NUMBER TO RETURN
005C    ;      TO THE CALLING TRANSITION ROUTINE.
005C    ;
005C
005C           $STATE  EXPRESSION
005C           $TRAN   !TERM
005C           $STATE  EXPOP
005C           $TRAN   '+',ADDITION
005C           $TRAN   '-',SUBTRACTION
005C           $TRAN   TPA$_LAMBDA,TPA$_EXIT
005C           $STATE  ADDITION
005C           $TRAN   !EXPRESSION,TPA$_EXIT,ADD2
005C           $STATE  SUBTRACTION
005C           $TRAN   !TERM, EXPOP, SUB2
005C    ;      $TRAN   !EXPRESSION,TPA$_EXIT,SUB2
005C
005C           $STATE  TERM
005C           $TRAN   !FACTOR
005C           $STATE
005C           $TRAN   '@',ARITH_SHIFT
005C           $TRAN   '*',MULTIPLICATION
005C           $TRAN   !DIVISION,TPA$_EXIT
005C           $TRAN   TPA$_LAMBDA,TPA$_EXIT
005C           $STATE  ARITH_SHIFT
005C           $TRAN   !TERM,TPA$_EXIT,SHIFT2
```

```
005C          $STATE  MULTIPLICATION
005C          $TRAN   !TERM,TPA$_EXIT,MULT2
005C          $STATE  DIVISION
005C          $TRAN   '/'
005C          $STATE
005C          $TRAN   !TERM,TPA$_EXIT,DIV2
005C
005C          $STATE  FACTOR
005C          $TRAN   '-',NEGATE
005C          $TRAN   '(',PARENS
005C          $TRAN   '@',INDIRECT
005C          $TRAN   '.',TPA$_EXIT,EVAL_DOT
005C          $TRAN   TPA$_SYMBOL,TPA$_EXIT,EVAL_SYMBOL
005C          $TRAN   !CONSTANT,TPA$_EXIT,EVAL_CONSTANT
005C          $STATE  NEGATE
005C          $TRAN   !FACTOR,TPA$_EXIT,NEG1
005C          $STATE  INDIRECT
005C          $TRAN   !FACTOR,TPA$_EXIT,EVAL_CONTENTS
005C          $STATE  PARENS
005C          $TRAN   !EXPRESSION
005C          $STATE
005C          $TRAN   ')',TPA$_EXIT
005C
005C          $STATE  CONSTANT
005C          $TRAN   '+',CONSTANT
005C          $TRAN   '^',RADIX
005C          $TRAN   !HEX,TPA$_EXIT
005C
005C          $STATE  RADIX
005C          $TRAN   'O',OCTAL
005C          $TRAN   'X',HEX
005C          $TRAN   'D',DECIMAL
005C
005C          $STATE  OCTAL
005C          $TRAN   TPA$_OCTAL,TPA$_EXIT
005C
005C          $STATE  DECIMAL
005C          $TRAN   TFA$_DECIMAL,TPA$_EXIT
005C
005C          $STATE  HEX
005C          $TRAN   'G',SYSREG
005C          $TRAN   'H',CTLREG
005C          $TRAN   TPA$_HEX,TPA$_EXIT
005C          $STATE  SYSREG
005C          $TRAN   TPA$_HEX,TPA$_EXIT,ADD8000
005C          $STATE  CTLREG
005C          $TRAN   TPA$_HEX,TPA$_EXIT,ADD7FFE
005C
005C      ;
005C      ;       SHOW PROCESS [name/ALL] [/INDEX=n] [/SYSTEM]
005C      ;               /WORKING_SET_LIST                       [or WSL]
005C      ;               /PROCESS_SECTION_TABLE                  [or PST]
005C      ;               /PAGE_TABLES                            [or PPT]
005C      ;                       start:end
005C      ;                       start;length
005C      ;                       /P0
005C      ;                       /P1
```

```
005C  ;               /REGISTERS
005C  ;               /PCB
005C  ;               /PHD
005C  ;               /ALL
005C  ;               /RMS [=opt]
005C  ;                   [=(opt1,opt2,...,optn)]
005C  ;               /LOCKS
005C  ;               /CHANNELS
005C  ;
005C
005C          $STATE  PROCESS
005C          $TRAN   !CK_EOS,TPA$_EXIT,SHOW_PROCESS
005C          $TRAN   '/'
005C          $TRAN   TPA$_SYMBOL,PROCESS,,,PROC_NAME
005C          $STATE  PROCESS1
005C          $TRAN   'INDEX',PROC_PIX
005C          $TRAN   'SYSTEM',PROCESS,,OPT$M_SYSPROC,OPTIONS
005C          $TRAN   'WORKING_SET_LIST',PROCESS,,OPT$M_WSL,OPTIONS
005C          $TRAN   'WSL',PROCESS,,OPT$M_WSL,OPTIONS
005C          $TRAN   'PROCESS_SECTION_TABLE',PROCESS,,OPT$M_PST,OPTIONS
005C          $TRAN   'PST',PROCESS,,OPT$M_PST,OPTIONS
005C          $TRAN   'PAGE_TABLES',PROC_PPT,,OPT$M_PPT,OPTIONS
005C          $TRAN   'PPT',PROC_PPT,,OPT$M_PPT,OPTIONS
005C          $TRAN   'P0',PROCESS,,<OPT$M_PPT!OPT$M_P0_PPT>,OPTIONS
005C          $TRAN   'P1',PROCESS,,<OPT$M_PPT!OPT$M_P1_PPT>,OPTIONS
005C          $TRAN   'REGISTERS',PROCESS,,OPT$M_REGS,OPTIONS
005C          $TRAN   'PCB',PROCESS,,OPT$M_PCB,OPTIONS
005C          $TRAN   'PHD',PROCESS,,OPT$M_PHD,OPTIONS
005C          $TRAN   'ALL',PROCESS,,<OPT$M_PCB!OPT$M_PHD!OPT$M_REGS!-
005C  OPT$M_WSL!OPT$M_PST!OPT$M_PPT!OPT$M_CHAN>,OPTIONS
005C          $TRAN   'LOCKS',PROCESS,,OPT$M_LCK,OPTIONS
005C          $TRAN   'CHANNELS',PROCESS,,OPT$M_CHAN,OPTIONS
005C          $TRAN   !RMS,PROCESS,,OPT$M_RMS,OPTIONS
005C          $STATE  PROC_PIX
005C          $TRAN   ':'
005C          $TRAN   '='
005C          $STATE
005C          $TRAN   TPA$_HEX,PROCESS,STORE_PROC_INDEX
005C          $STATE  PROC_PPT
005C          $TRAN   !CK_EOS,TPA$_EXIT,SHOW_PROCESS
005C          $TRAN   '/',PROCESS1
005C          $TRAN   !EXPRESSION
005C          $STATE
005C          $TRAN   SEMI,,,OPT$M_PPT_LEN,OPTIONS
005C          $TRAN   COLON,,,OPT$M_PPT_RNG,OPTIONS
005C          $STATE
005C          $TRAN   !EXPRESSION,PROCESS
005C
005C  ;
005C  ;       SHOW PROCESS/RMS [=opt]
005C  ;                        [=(opt1,opt2,...,optn)]
005C  ;
005C
005C          $STATE  RMS
005C           $TRAN  'RMS'
005C          $STATE
005C           $TRAN  !DISPLAY,TPA$_EXIT,,OPT$M_RMSD,OPTIONS
```

```
005C                   $TRAN   TPA$_LAMBDA,TPA$_EXIT
005C
005C          ;
005C          ;       SET PROCESS name /INDEX=n /SYSTEM
005C          ;
005C
005C                  $STATE  SET_PROC
005C                  $TRAN   !INDEX
005C                  $TRAN   TPA$_SYMBOL,,,,PROC_NAME
005C                  $STATE
005C                  $TRAN   TPA$_LAMBDA,TPA$_EXIT,SET_PROCESS
005C
005C                  $STATE  INDEX
005C                  $TRAN   '/'
005C                  $STATE
005C                  $TRAN   'INDEX'
005C                  $TRAN   'SYSTEM',TPA$_EXIT,,OPT$M_SYSPROC,OPTIONS
005C                  $STATE
005C                  $TRAN   ':'
005C                  $TRAN   '='
005C                  $STATE
005C                  $TRAN   TPA$_HEX,TPA$_EXIT,STORE_PROC_INDEX
005C
005C          ;
005C          ;       SHOW POOL /IRP /NONPAGED /PAGED /ALL /SUMMARY /HEADER
005C          ;
005C
005C                  $STATE  POOL
005C                  $TRAN   !CK_EOS,TPA$_EXIT,SHOW_POOL
005C                  $TRAN   '/'
005C                  $TRAN   TPA$_LAMBDA,POOL_RANGE
005C                  $STATE
005C                  $TRAN   'IRP',POOL,,OPT$M_IRP,OPTIONS
005C                  $TRAN   'LRP',POOL,,OPT$M_LRP,OPTIONS
005C                  $TRAN   'SRP',POOL,,OPT$M_SRP,OPTIONS
005C                  $TRAN   'NONPAGED',POOL,,OPT$M_NONPAGED,OPTIONS
005C                  $TRAN   'PAGED',POOL,,OPT$M_PAGED,OPTIONS
005C                  $TRAN   'ALL',POOL,,OPT$M_IRP!OPT$M_NONPAGED!OPT$M_PAGED,OPTIONS
005C                  $TRAN   'SUMMARY',POOL,,OPT$M_SUMMARY,OPTIONS
005C                  $TRAN   'HEADER',POOL,,OPT$M_HEADER,OPTIONS
005C                  $TRAN   'FREE',POOL,,OPT$M_FREE,OPTIONS
005C                  $TRAN   'TYPE',,,OPT$M_TYPE,OPTIONS
005C                  $STATE
005C                  $TRAN   '='
005C                  $TRAN   ':'
005C                  $STATE
005C                  $TRAN   TPA$_SYMBOL,POOL,,,STRUCTURE
005C
005C          ;
005C          ;       SHOW POOL  start:end
005C          ;                  start;length
005C          ;
005C
005C                  $STATE  POOL_RANGE
005C                  $TRAN   !EXPRESSION
005C                  $STATE
005C                  $TRAN   SEMI,,,OPT$M_LENGTH,OPTIONS
```

```
005C                  $TRAN   COLON,,,OPT$M_RANGE,OPTIONS
005C                  $STATE
005C                  $TRAN   !EXPRESSION
005C                  $STATE
005C                  $TRAN   !CK_EOS,TPA$_EXIT,SHOW_POOL_RANGE
005C
005C           ;
005C           ;      SHOW STACK /mode
005C           ;
005C
005C                  $STATE  STACK
005C                  $TRAN   !CK_EOS,TPA$_EXIT,PRINT_STACKS
005C                  $TRAN   '/'
005C                  $TRAN   TPA$_LAMBDA,STACK_RANGE
005C                  $STATE
005C                  $TRAN   'INTERRUPT',STACK,,OPT$M_ISP,OPTIONS
005C                  $TRAN   'KERNEL',STACK,,OPT$M_KSP,OPTIONS
005C                  $TRAN   'EXECUTIVE',STACK,,OPT$M_ESP,OPTIONS
005C                  $TRAN   'SUPERVISOR',STACK,,OPT$M_SSP,OPTIONS
005C                  $TRAN   'USER',STACK,,OPT$M_USP,OPTIONS
005C                  $TRAN   'ALL',STACK,,OPT$M_ALL,OPTIONS
005C
005C           ;
005C           ;      SHOW STACK start:end
005C           ;                 start;length
005C           ;
005C
005C                  $STATE  STACK_RANGE
005C                  $TRAN   !EXPRESSION
005C                  $STATE
005C                  $TRAN   SEMI,,,OPT$M_LENGTH,OPTIONS
005C                  $TRAN   COLON,,,OPT$M_RANGE,OPTIONS
005C                  $STATE
005C                  $TRAN   !EXPRESSION
005C                  $STATE
005C                  $TRAN   !CK_EOS,TPA$_EXIT,PRINT_ANY_STACK
005C
005C           ;
005C           ;      SHOW PFN_DATA <optional pfn>
005C           ;
005C
005C                  $STATE  PFN_DATA
005C                  $TRAN   '/',PFN_OPTION
005C                  $TRAN   !EXPRESSION,,,OPT$M_SINGLEPFN,OPTIONS
005C                  $TRAN   TPA$_LAMBDA,,,OPT$M_FREE!OPT$M_MODIFIED!OPT$M_BAD!OPT$M_WHOLEPFN,OPT
005C                  $STATE
005C                  $TRAN   TPA$_LAMBDA,TPA$_EXIT,DISPLAY_PFN
005C
005C                  $STATE  PFN_OPTS
005C                  $TRAN   '/'
005C                  $TRAN   TPA$_LAMBDA,TPA$_EXIT,DISPLAY_PFN
005C                  $STATE  PFN_OPTION
005C                  $TRAN   'FREE',PFN_OPTS,,OPT$M_FREE,OPTIONS
005C                  $TRAN   'MODIFIED',PFN_OPTS,,OPT$M_MODIFIED,OPTIONS
005C                  $TRAN   'BAD',PFN_OPTS,,OPT$M_BAD,OPTIONS
005C                  $TRAN   'SYSTEM',PFN_OPTS,,OPT$M_WHOLEPFN,OPTIONS
005C                  $TRAN   'ALL',PFN_OPTS,,OPT$M_FREE!OPT$M_MODIFIED!OPT$M_BAD!OPT$M_WHOLEPFN,O
```

```
005C
005C   ;
005C   ;       SHOW PAGE_TABLE/SYSTEM/GLOBAL/ALL
005C   ;
005C
005C           $STATE  PAGE_TABLE
005C           $TRAN   '/',PAGE_OPTION
005C           $TRAN   TPA$_LAMBDA,,,OPT$M_SYSTEM!OPT$M_GLOBAL,OPTIONS
005C           $STATE
005C           $TRAN   !CK_EOS,TPA$_EXIT,DISPLAY_SPT
005C           $TRAN   TPA$_LAMBDA,PAGE_RANGE
005C
005C           $STATE  PAGE_OPTS
005C           $TRAN   '/'
005C           $TRAN   !CK_EOS,TPA$_EXIT,DISPLAY_SPT
005C           $TRAN   TPA$_LAMBDA,PAGE_RANGE
005C           $STATE  PAGE_OPTION
005C           $TRAN   'SYSTEM',PAGE_OPTS,,OPT$M_SYSTEM,OPTIONS
005C           $TRAN   'GLOBAL',PAGE_OPTS,,OPT$M_GLOBAL,OPTIONS
005C           $TRAN   'ALL',PAGE_OPTS,,OPT$M_ALL,OPTIONS
005C
005C
005C   ;
005C   ;       SHOW PAGE  start:end
005C   ;                  start;length
005C   ;
005C
005C           $STATE  PAGE_RANGE
005C           $TRAN   !EXPRESSION
005C           $STATE
005C           $TRAN   SEMI,,,OPT$M_LENGTH,OPTIONS
005C           $TRAN   COLON,,,OPT$M_RANGE,OPTIONS
005C           $STATE
005C           $TRAN   !EXPRESSION
005C           $STATE
005C           $TRAN   !CK_EOS,TPA$_EXIT,DISPLAY_SPT_RANGE
005C
005C   ;
005C   ;       SHOW SUMMARY /IMAGE
005C   ;
005C
005C           $STATE  SHOW_SUMMARY
005C           $TRAN   !CK_EOS,TPA$_EXIT,PROCESS_SUMMARY
005C           $TRAN   '/'
005C           $STATE
005C           $TRAN   'IMAGE',SHOW_SUMMARY,,OPT$M_IMAGE,OPTIONS
005C
005C   ;
005C   ;       SHOW SYMBOL[/ALL] [name] [/ALL]
005C   ;
005C
005C           $STATE  SYMBOL
005C           $TRAN   TPA$_LAMBDA,,CLEAR_SYMBOL_NAME
005C           $STATE  SYMBOL_QUALS
005C           $TRAN   '/',SYMBOL_OPTS
005C           $TRAN   TPA$_SYMBOL,SYM,,,SYMBOL_NAME
005C           $TRAN   !CK_EOS,TPA$_EXIT,SHOW_SYMBOL
```

```
005C          $STATE  SYM
005C          $TRAN   '/'
005C          $TRAN   !CK_EOS,TPA$_EXIT,SHOW_SYMBOL
005C          $STATE  SYMBOL_OPTS
005C          $TRAN   'ALL',SYMBOL_QUALS,,OPT$M_ALL,OPTIONS
005C
005C    ;
005C    ;     SHOW DEVICE device
005C    ;
005C
005C          $STATE  DEVICE
005C          $TRAN   '/',DEVICE_OPTS
005C          $TRAN   TPA$_SYMBOL,TPA$_EXIT,DISPLAY_DEVICE
005C          $TRAN   TPA$_LAMBDA,TPA$_EXIT,DISPLAY_DEVICE
005C
005C          $STATE  DEVICE_OPTS
005C          $TRAN   'ADDRESS'
005C          $STATE
005C          $TRAN   '='
005C          $STATE
005C          $TRAN   !EXPRESSION
005C          $STATE
005C          $TRAN   TPA$_LAMBDA,TPA$_EXIT,DISPLAY_DEVBYADDR
005C
005C    ;
005C    ;     DEFINE symbol = value
005C    ;
005C
005C          $STATE  DEFINE
005C          $TRAN   '/',TPA$_EXIT,DEFINE_KEY
005C          $TRAN   TPA$_SYMBOL,,,,SYMBOL_DESC
005C          $STATE
005C          $TRAN   '='
005C          $TRAN   TPA$_LAMBDA
005C          $STATE
005C          $TRAN   !EXPRESSION
005C          $STATE
005C          $TRAN   TPA$_LAMBDA,TPA$_EXIT,DEFINE_SYMBOL
005C
005C    ;
005C    ;     HELP COMMAND
005C    ;
005C
005C          $STATE  HELP_CMD
005C          $TRAN   !CK_EOS,TPA$_EXIT,DISPLAY_HELP
005C          $TRAN   !COLLECT,TPA$_EXIT,DISPLAY_HELP
005C
005C    ;
005C    ;     EXIT COMMAND
005C    ;
005C
005C          $STATE  EXIT_CMD
005C          $TRAN   !CK_EOS,TPA$_EXIT,EXIT_COMMAND
005C          $TRAN   TPA$_STRING,TPA$_EXIT,EXIT_COMMAND
005C
005C    ;
005C    ;     READ filespec
```

```
PAR
Sym

$$$
$$$
$$$
$$$
$$$
$$$
$$$
$$K
ADD
ADD
ADD
ADD
ADD
ARG
ARI
ATM
CDT
CDT
CK
CLE
CLO
CLR
COL
COL
COL
CON
COP
CSI
CTL
DEC
DEF
DEF
DEF
DEV
DEV
DIS
DIS
DIS
DIS
DIS
DIS
DIS
DIS
DIS
DIS
DIV
DIV
END
ESP
EVA
EVA
EVA
EVA
EVA
EVA
EVA
EVA
```

```
005C   ;
005C
005C          $STATE  READ_CMD
005C          $TRAN   !READ_QUALS
005C          $TRAN   TPA$_LAMBDA
005C          $STATE
005C          $TRAN   !COLLECT,,,,FILE_DESC
005C          $STATE
005C          $TRAN   TPA$_LAMBDA,TPA$_EXIT,READ_SYMFILE
005C
005C          $STATE  READ_QUALS
005C          $TRAN   '/'
005C          $STATE
005C          $TRAN   'RELOCATE'
005C          $STATE
005C          $TRAN   '='
005C          $STATE
005C          $TRAN   !EXPRESSION,TPA$_EXIT,STORE_RELOCATION
005C
005C   ;
005C   ;      COPY filespec
005C   ;
005C
005C          $STATE  COPY_CMD
005C          $TRAN   !COLLECT,,,,FILE_DESC
005C          $STATE
005C          $TRAN   TPA$_LAMBDA,TPA$_EXIT,SAVE_DUMP
005C
005C   ;
005C   ;      FORMAT address /TYPE=struct
005C   ;
005C
005C          $STATE  FORMAT_CMD
005C          $TRAN   !FORMAT_OPTS
005C          $STATE
005C          $TRAN   !EXPRESSION
005C          $STATE
005C          $TRAN   !FORMAT_OPTS
005C          $STATE
005C          $TRAN   TPA$_LAMBDA,TPA$_EXIT,FORMAT
005C
005C          $STATE  FORMAT_OPTS
005C          $TRAN   '/'
005C          $TRAN   TPA$_LAMBDA,TPA$_EXIT
005C          $STATE
005C          $TRAN   'TYPE'
005C          $STATE
005C          $TRAN   '='
005C          $TRAN   ':'
005C          $STATE
005C          $TRAN   TPA$_SYMBOL,,,,STRUCTURE
005C          $STATE
005C          $TRAN   TPA$_LAMBDA,TPA$_EXIT,,1,OPTIONS
005C
005C
005C          $END_STATE
005C
```

```
                         00000000   989          .PSECT  PARSE,EXE,NOWRT
```

```
SET
SET
SHI
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SHO
SMG
SMG
SMG
SNA
STA
STA
STA
STO
STO
STO
STO
STO
STO
STR
SUB
SUB
SYM
SYM
SYM
SYM
SYM
SYM
SYM
SYS
TER
TPA
TPA
TPA
TPA
TPA
TPA
TPA
TPA
TPA
TPA
TPA
```

```
                                 0000           .SBTTL  EXPRESSION ANALYSIS ACTION ROUTINES
                                 0000  ;---
                                 0000  ;
                                 0000  ;        EXPRESSION ANALYSIS ACTION ROUTINES
                                 0000  ;
                                 0000  ;        THESE ROUTINES ARE CALLED FROM THE EXPRESSION ANALYSIS
                                 0000  ;        PARSING TRANSITIONS TO EVALUATE ARGUMENTS USING A SEPARATE
                                 0000  ;        STACK SPACE.
                                 0000  ;
                                 0000  ;    INPUTS:
                                 0000  ;
                                 0000  ;        AP = ADDRESS OF TPARSE PARAMETER BLOCK
                                 0000  ;
                                 0000  ;---
                                 0000
                         0000    0000  ADD2:    .WORD   0
                      44   10    0002           BSBB    EVAL_SETUP
                 61   81   C0    0004           ADDL2   (R1)+,(R1)              ; ADD 2 ARGUMENTS, RESULT ON TOP
                      47   11    0007           BRB     STORE
                                 0009
                         0000    0009  MULT2:   .WORD   0
                      3B   10    000B           BSBB    EVAL_SETUP
                 61   81   C4    000D           MULL2   (R1)+,(R1)              ; MULTIPLY 2 ARGUMENTS, RESULT ON TOP
                      3E   11    0010           BRB     STORE
                                 0012
                         0000    0012  SUB2:    .WORD   0
                      32   10    0014           BSBB    EVAL_SETUP
                 61   81   C2    0016           SUBL2   (R1)+,(R1)              ; SUBTRACT 2 ARGUMENTS, RESULT ON TOP
                      35   11    0019           BRB     STORE
                                 001B
                         0000    001B  DIV2:    .WORD   0
                      29   10    001D           BSBB    EVAL_SETUP
                      61   D5    001F           TSTL    (R1)                    ; DO NOT ALLOW DIVIDE BY ZERO
                      3C   13    0021           BEQL    FAILURE
                 61   81   C6    0023           DIVL2   (R1)+,(R1)              ; DIVIDE 2 ARGUMENTS, RESULT ON TOP
                      28   11    0026           BRB     STORE
                                 0028
                         0000    0028  SHIFT2:  .WORD   0
                      1C   10    002A           BSBB    EVAL_SETUP
                 50   81   D0    002C           MOVL    (R1)+,R0                ; GET SHIFT COUNT
            61   61   50   78    002F           ASHL    R0,(R1),(R1)            ; SHIFT LEFT BY COUNT
                      1B   11    0033           BRB     STORE
                                 0035
                         0000    0035  NEG1:    .WORD   0
                      0F   10    0037           BSBB    EVAL_SETUP
                 61   61   CE    0039           MNEGL   (R1),(R1)               ; NEGATE 1 ARGUMENT, RESULT ON TOP
                      12   11    003C           BRB     STORE
                                 003E
                                 003E  EVAL_CONSTANT:
                         0000    003E           .WORD   0
                      06   10    0040           BSBB    EVAL_SETUP
             71    1C AC   D0    0042           MOVL    TPA$L_NUMBER(AP),-(R1)  ; PUSH RESULT FROM TPARSE
                      08   11    0046           BRB     STORE
                                 0048
                                 0048  EVAL_SETUP:
       51    00000000'EF   D0    0048           MOVL    ESP,R1                  ; R1 = STACK POINTER
                           05    004F           RSB
```

```
PARS
Pse

PSE
----
.
$AB
SDA
_LI
_LI
_LI
PAR

Pha
----
Ini
Com
Pas
Sym
Pas
Sym
Pse
Cro
Ass

The
519
The
137
32

Mac
----
_$2
_$2
_$2
TOT

753

The

MAC
```

```
                                 0050  1048
                                 0050  1049 STORE:
              1C AC    61    D0  0050  1050            MOVL    (R1),TPA$L_NUMBER(AP)       ; SAVE RESULT FOR CALLER
        00000000'EF    51    D0  0054  1051            MOVL    R1,ESP                      ; STORE STACK POINTER
                 50    01    D0  005B  1052            MOVL    #1,R0
                             04  005E  1053            RET
                                 005F  1054
                                 005F  1055 FAILURE:
                       50    D4  005F  1056            CLRL    R0
                             04  0061  1057            RET
                                 0062  1058
                                 0062  1059 EVAL_SYMBOL:
                           0000  0062  1060            .WORD   0
              10 AC    7F        0064  1061            PUSHAQ  TPA$L_TOKENCNT(AP)          ; DESCRIPTOR OF SYMBOL NAME
        00000000'EF    01    FB  0067  1062            CALLS   #1,SYMBOL_VALUE             ; TRY TO FIND IN SYMBOL TABLE
                 EE    50    E9  006E  1063            BLBC    R0,FAILURE
                             51  DD  0071  1064        PUSHL   R1
                       D3    10  0073  1065            BSBB    EVAL_SETUP
                       71  8ED0  0075  1066            POPL    -(R1)                       ; PUSH RESULT ONTO STACK
                       D6    11  0078  1067            BRB     STORE
                                 007A  1068
                                 007A  1069 EVAL_DOT:
                           0000  007A  1070            .WORD   0
                       CA    10  007C  1071            BSBB    EVAL_SETUP
      71    00000000'EF      D0  007E  1072            MOVL    ADDRESS,-(R1)
                       C9    11  0085  1073            BRB     STORE
                                 0087  1074
                                 0087  1075 EVAL_CONTENTS:
                           0000  0087  1076            .WORD   0
                       BD    10  0089  1077            BSBB    EVAL_SETUP
                                 008B  1078            REQMEM  @(R1)                       ; GET CONTENTS OF LOCATION
                       51    DD  0095  1079            PUSHL   R1
                       AF    10  0097  1080            BSBB    EVAL_SETUP
                       61  8ED0  0099  1081            POPL    (R1)                        ; AND REPLACE ARG. ON TOP OF STACK
                       B2    11  009C  1082            BRB     STORE
                                 009E  1083
                                 009E  1084 ADD8000:                                       ; ADDRESS IN SYSTEM REGION
                           0000  009E  1085            .WORD   0
   1C AC    80000000 8F      C0  00A0  1086            ADDL    #^X80000000,TPA$L_NUMBER(AP) ; ADD BASE OF SYSTEM REGION
                             04  00A8  1087            RET
                                 00A9  1088
                                 00A9  1089 ADD7FFE:                                       ; ADDRESS IN CONTROL REGION
                           0000  00A9  1090            .WORD   0
   1C AC    7FFE0000 8F      C0  00AB  1091            ADDL    #^X7FFE0000,TPA$L_NUMBER(AP) ; CONTROL REGION
                             04  00B3  1092            RET
```

**F

```
                                00B4            .SBTTL  STORE_RELOCATION, STORE READ/RELOCATE VALUE
                                00B4    ;---
                                00B4    ;
                                00B4    ;       STORE THE VALUE OF THE READ/RELOCATE QUALIFIER.
                                00B4    ;
                                00B4    ; INPUTS:
                                00B4    ;
                                00B4    ;       TPA$_L_NUMBER(AP) = RELOCATION BASE
                                00B4    ;
                                00B4    ; OUTPUTS:
                                00B4    ;
                                00B4    ;       RELOCATE_BASE = RELOCATION BASE
                                00B4    ;---
                                00B4
                                00B4    STORE_RELOCATION:
                          0000  00B4            .WORD   0
                                00B6
     00000000'EF    1C AC   D0  00B6            MOVL    TPA$L_NUMBER(AP),RELOCATE_BASE
                            04  00BE            RET
                                00BF
                                00BF            .SBTTL  STORE_PROC_INDEX, STORE PROCESS INDEX LOW BITS
                                00BF    ;---
                                00BF    ;
                                00BF    ;       STORE THE LOW ORDER BITS (AS DETERMINED BY PIX_WIDTH) OF THE PROCESS
                                00BF    ;       INDEX.  THIS LETS US INTERCHANGE INTERNAL AND EXTENDED PIDS
                                00BF    ;
                                00BF    ; INPUTS:
                                00BF    ;
                                00BF    ;       TPA$_L_NUMBER(AP) = USER-SPECIFIED PROC INDEX
                                00BF    ;
                                00BF    ; OUTPUTS:
                                00BF    ;
                                00BF    ;       PROC_INDEX = LOW ORDER BITS
                                00BF    ;---
                                00BF
                                00BF    STORE_PROC_INDEX:
                          0000  00BF            .WORD   0
                                00C1
   1C AC    00000000'EF  00   EF  00C1            EXTZV   #0, PIX_WIDTH, TPA$L_NUMBER(AP), PROC_INDEX
                 00000000'EF      00CA
                     50  01   D0  00CF            MOVL    #1, R0
                                00D2
                            04  00D2            RET
```

```
                                   00D3          .SBTTL  SAVE_COMMAND, SAVE THE CURRENT COMMAND
                                   00D3    ;---
                                   00D3    ;
                                   00D3    ;        SAVE THE CURRENT COMMAND IN THE COMMAND BUFFER
                                   00D3    ;        SO THAT A LATER 'REPEAT' COMMAND CAN RECALL IT.
                                   00D3    ;
                                   00D3    ; INPUTS:
                                   00D3    ;
                                   00D3    ;        TPA$L_STRINGCNT(AP) = DESCRIPTOR OF STRING
                                   00D3    ;
                                   00D3    ; OUTPUTS:
                                   00D3    ;
                                   00D3    ;        PREV_COMMAND = DESCRIPTOR OF SAVED STRING
                                   00D3    ;
                                   00D3    ;---
                                   00D3
                                   00D3    SAVE_COMMAND::
                            003C   00D3          .WORD   ^M<R2,R3,R4,R5>
                                   00D5
    00000050'EF    08 AC      B0   00D5          MOVW    TPA$L_STRINGCNT(AP),PREV_COMMAND
          0C BC    08 AC      28   00DD          MOVC    TPA$L_STRINGCNT(AP),@TPA$L_STRINGPTR(AP),-
       00000054'FF                 00E2                  @PREV_COMMAND+4
                                   00E7          ;
                                   00E7          ; Tell the SMG$ level how to repeat this command
                                   00E7          ;
                 7E           D4   00E7          CLRL    -(SP)                     ; State string null
       00000050'EF            9F   00E9          PUSHAB  PREV_COMMAND              ; Last command
       00000058'EF            9F   00EF          PUSHAB  KEY_ATTR                  ; attributes
                 7E           D4   00F5          CLRL    -(SP)                     ; No IF_STATE
       00000000'EF            9F   00F7          PUSHAB  REPEAT_KEY                ; Repeat key name (KP0 defaults)
       00000000'EF            9F   00FD          PUSHAB  KEYTABLE                  ; key table
  00000000'GF      06         FB   0103          CALLS   #6,G^SMG$ADD_KEY_DEF      ; Change key definition
                                   010A          SIGNAL
                              04   0116          RET
                              0117          .SBTTL  REPEAT_COMMAND, REPEAT THE PREVIOUS COMMAND
                              0117     ;---
                              0117     ;
                              0117     ;        STORE THE PREVIOUS COMMAND IN THE COMMAND BUFFER
                              0117     ;        AND RESTART THE PARSE.
                              0117     ;
                              0117     ; INPUTS:
                              0117     ;
                              0117     ;        PREV_COMMAND = DESCRIPTOR OF PREVIOUS COMMAND
                              0117     ;
                              0117     ; OUTPUTS:
                              0117     ;
                              0117     ;        TPA$L_STRINGCNT(AP) = THE PREVIOUS COMMAND IS RELOADED
                              0117     ;
                              0117     ;---
                              0117
                              0117     REPEAT_COMMAND::
                        0000  0117              .WORD   0
                              0119
      08 AC   00000050'EF  7D  0119              MOVQ    PREV_COMMAND,TPA$L_STRINGCNT(AP)
                           04  0121              RET
```

```
                                  0122  1194          .SBTTL  DEFINE_KEY - DEFINE KEY
                                  0122  1195  ;+++
                                  0122  1196  ;
                                  0122  1197  ;       Use input buffer as input, calls SMG$ to parse DEF/KEY.
                                  0122  1198  ;
                                  0122  1199  ;---
                                  0122  1200
                            0004  0122  1201  .ENTRY  DEFINE_KEY,^M<R2>
                                  0124  1202
          00000000'EF   DF  0124  1203          PUSHAL  INPUT_BUFFER                    ; address
      7E  00000000'EF   3C  012A  1204          MOVZWL  INPUT_LEN,-(SP)                 ; length
                   5E   DD  0131  1205          PUSHL   SP                              ; address of desc
          00000000'EF   9F  0133  1206          PUSHAB  KEYTABLE                        ; table id
   00000000'GF    02    FB  0139  1207          CALLS   #2,G^SMG$DEFINE_KEY             ; Define key...
            5E    08    C0  0140  1208          ADDL    #8,SP                           ; pop
                                  0143  1209          SIGNAL
                                  014F  1210          STATUS  SUCCESS                         ; ok
                            04    0156  1211          RET
                                  0157  1212
```

```
                                     0157          .SBTTL  Action Routines for SET!SHOW RMS/DISPLAY=optionspec
                                     0157  ;+++
                                     0157  ; SETUP_RMS_TMP -- Setup the temporary RMS options word.
                                     0157  ;
                                     0157  ; Inputs:
                                     0157  ;
                                     0157  ;         None.
                                     0157  ;
                                     0157  ; Outputs:
                                     0157  ;
                                     0157  ;         RMS_DIS_TMP = 0
                                     0157  ;         RMS_IFI_TMP = 0
                                     0157  ;---
                                     0157
                               0000  0157  SETUP_RMS_TMP:  .WORD   0
                00000000'EF      D4  0159          CLRL    RMS_DIS_TMP
                00000000'EF      B4  015F          CLRW    RMS_IFI_TMP
                         50  01  D0  0165          MOVL    #1,R0
                                 04  0168          RET
                                     0169
                                     0169  ;+++
                                     0169  ; STORE_TMP1_OPT -- Merge this particular option into temporary option word.
                                     0169  ;
                                     0169  ; Inputs:
                                     0169  ;
                                     0169  ;         RMS_DIS_TMP1 = Outcome of !DIS_OPT state (Possibly containing OPT$M_NO)
                                     0169  ;
                                     0169  ; Outputs:
                                     0169  ;
                                     0169  ;         RMS_DIS_TMP is set or cleared with value of RMS_DIS_TMP1 depending
                                     0169  ;                 on value of OPT$M_NO in RMS_DIS_TMPT.
                                     0169  ;---
                                     0169
                               0000  0169  STORE_TMP1_OPT: .WORD   0
           00000000'EF       01  D3  016B          BITL    #OPT$M_NO,RMS_DIS_TMP1  ; was this display option ''no'ed''?
                             0F  12  0172          BNEQ    10$                     ; if neq yes
  00000000'EF    00000000'EF     C8  0174          BISL    RMS_DIS_TMP1,RMS_DIS_TMP
                         50  01  D0  017F          MOVL    #1,R0
                                 04  0182          RET
  00000000'EF    00000000'EF     CA  0183  10$:    BICL    RMS_DIS_TMP1,RMS_DIS_TMP
                         50  01  D0  018E          MOVL    #1,R0
                                 04  0191          RET
                                     0192
                                     0192  ;+++
                                     0192  ; END_SET_RMS -- Finish the SET RMS Command.
                                     0192  ;
                                     0192  ; Inputs:
                                     0192  ;
                                     0192  ;         Final outputs of the SET RMS command:
                                     0192  ;                 RMS_DIS_TMP
                                     0192  ;                 RMS_IFI_TMP
                                     0192  ;
                                     0192  ; Outputs:
                                     0192  ;
                                     0192  ;         Permanent display options set:
                                     0192  ;                 RMS_DIS_OPT = RMS_DIS_TMP
                                     0192  ;                 RMS_IFI     = RMS_IFI_TMP
```

```
                                  0192  ;---
                                  0192
                           0000   0192  END_SET_RMS:.WORD      0
00000000'EF   00000000'EF    D0   0194          MOVL    RMS_DIS_TMP,RMS_DIS_OPT
00000000'EF   00000000'EF    B0   019F          MOVW    RMS_IFI_TMP,RMS_IFI
                   50   01   D0   01AA          MOVL    #1,R0
                             04   01AD          RET
                                  01AE
                                  01AE  ;+++
                                  01AE  ; STORE_IFI_TMP -- Set the temporary IFI from the results of !EXPRESSION.
                                  01AE  ;
                                  01AE  ; Inputs:
                                  01AE  ;
                                  01AE  ;         TPA$L_NUMBER(AP) = Results of !EXPRESSION
                                  01AE  ;
                                  01AE  ; Outputs:
                                  01AE  ;
                                  01AE  ;         RMS_IFI_TMP contains above value.
                                  01AE  ;---
                                  01AE
                           0000   01AE  STORE_IFI_TMP:.WORD    0
     00000000'EF    1C AC    B0   01B0          MOVW    TPA$L_NUMBER(AP),RMS_IFI_TMP
                   50   01   D0   01B8          MOVL    #1,R0
                             04   01BB          RET
                                  01BC
                                  01BC  ;+++
                                  01BC  ; CLR_IFI_TMP -- Clear RMS_IFI_TMP.
                                  01BC  ;
                                  01BC  ; Inputs:
                                  01BC  ;
                                  01BC  ;         None.
                                  01BC  ;
                                  01BC  ; Outputs:
                                  01BC  ;
                                  01BC  ;         RMS_IFI_TMP = 0
                                  01BC  ;---
                                  01BC
                           0000   01BC  CLR_IFI_TMP:.WORD 0
              00000000'EF    B4   01BE          CLRW    RMS_IFI_TMP
                   50   01   D0   01C4          MOVL    #1,R0
                             04   01C7          RET
                                  01C8
                                  01C8  ;+++
                                  01C8  ; SETUP_DIS_OPT -- Setup for !DIS_OPT state.
                                  01C8  ;
                                  01C8  ; Inputs:
                                  01C8  ;
                                  01C8  ;         None.
                                  01C8  ;
                                  01C8  ; Outputs:
                                  01C8  ;
                                  01C8  ;         RMS_DIS_TMP1 = 0
                                  01C8  ;---
                                  01C8
                           0000   01C8  SETUP_DIS_OPT:.WORD    0
              00000000'EF    D4   01CA          CLRL    RMS_DIS_TMP1
                   50   01   D0   01D0          MOVL    #1,R0
```

```
                              04  01D3   1328            RET
                                  01D4   1329
                                  01D4   1330   ;+++
                                  01D4   1331   ; STORE_STAR -- Include into temporary options previous perm options.
                                  01D4   1332   ;
                                  01D4   1333   ; Inputs:
                                  01D4   1334   ;
                                  01D4   1335   ;        RMS_DIS_OPT = Permanent options.
                                  01D4   1336   ;        RMS_IFI = Permanent IFI.
                                  01D4   1337   ;
                                  01D4   1338   ; Outputs:
                                  01D4   1339   ;
                                  01D4   1340   ;        RMS_DIS_TMP1 = RMS_DIS_OPT - OPT$M_NO
                                  01D4   1341   ;        RMS_IFI_TMP = RMS_IFI
                                  01D4   1342   ;---
                                  01D4   1343
                            0000  01D4   1344   STORE_STAR:.WORD 0
                 00000000'EF  DD  01D6   1345            PUSHL    RMS_DIS_OPT
                      6E  01  CA  01DC   1346            BICL     #OPT$M_NO,(SP)
                 00000000'EF 8ED0 01DF   1347            POPL     RMS_DIS_TMP1
      00000000'EF 00000000'EF  B0  01E6   1348            MOVW     RMS_IFI,RMS_IFI_TMP
                      50  01  D0  01F1   1349            MOVL     #1,R0
                              04  01F4   1350            RET
                                  01F5   1351   ;+++
                                  01F5   1352   ; CLEAR_SYMBOL_NAME -- Clear SYMBOL_NAME prior to parsing SHOW/SYMBOL
                                  01F5   1353   ;
                                  01F5   1354   ; Inputs:
                                  01F5   1355   ;
                                  01F5   1356   ;        None.
                                  01F5   1357   ;
                                  01F5   1358   ; Outputs:
                                  01F5   1359   ;
                                  01F5   1360   ;        SYMBOL_NAME = 0
                                  01F5   1361   ;---
                                  01F5   1362
                                  01F5   1363   CLEAR_SYMBOL_NAME:
                            0000  01F5   1364            .WORD 0
                 00000000'EF  7C  01F7   1365            CLRQ     SYMBOL_NAME
                      50  01  D0  01FD   1366            MOVL     #1,R0
                              04  0200   1367            RET
```

```
            0201  1369
            0201  1370         .END
```

```
$$$AST                  = 00000000                EXAMINE                   000003D0  R     03
$$$CNT                  = 00000003                EXAMINE_EXPRESSION        00000430  R     03
$$$FLG                  = FFFFFFFF                EXAM_INS                  000004B0  R     03
$$$KEY                  = 00000087                EXAM_MEMORY               *********    X  03
$$$KFG                  = FFFFFFFF                EXAM_OK                   000004A8  R     03
$$$MOD                  = 00000000                EXAM_QUALS                0000046C  R     03
$$$TMP                  = 00000398  R     05      EXAM_QUAL_LOOP            00000464  R     03
$$KEYTAB                = 00000000  R     04      EXIT_CMD                  000009E4  R     03
ADD2                      00000000  R     06      EXIT_COMMAND              *********    X  03
ADD7FFE                   000000A9  R     06      EXPOP                     00000504  R     03
ADD8000                   0000009E  R     06      EXPRESSION                00000500  R     03
ADDITION                  00000510  R     03      FACTOR                    0000055A  R     03
ADDRESS                   *********    X  06      FAILURE                   0000005F  R     06
ARGS                    = 00000001                FILE_DESC                 *********    X  03
ARITH_SHIFT               0000053A  R     03      FORMAT                    *********    X  03
ATM_OPT                   00000250  R     03      FORMAT_CMD                00000A2C  R     03
CDT_BYADDR                *********    X  03      FORMAT_OPTS               00000A40  R     03
CDT_SPCFY                 *********    X  03      HELP_CMD                  000009D0  R     03
CK_EOS                    00000046  R     03      HEX                       000005BE  R     03
CLEAR_SYMBOL_NAME         000001F5  R     06      IFI                       00000384  R     03
CLOSE_LOG                 *********    X  03      IFI2                      0000038C  R     03
CLR_IFI_TMP               000001BC  R     06      INDEX                     00000712  R     03
COLLECT                   000001EE  R     03      INDIRECT                  0000058A  R     03
COLLECT_LOOP              000001F0  R     03      INDIRECT_CMD              0000007A  R     03
COLON                   = 0000003A                INDIRECT_COMMAND          *********    X  03
CONSTANT                  0000059C  R     03      INPUT_BUFFER              *********    X  06
COPY_CMD                  00000A1C  R     03      INPUT_LEN                 *********    X  06
CSID                      *********    X  03      KEYTABLE                  *********    X  06
CTLREG                    000005D2  R     03      KEY_ATTR                  00000058  R     02
DECIMAL                   000005BA  R     03      LIB$SIGNAL                *********    X  06
DEFINE                    000009B2  R     03      LOCK                      000000D4  R     03
DEFINE_KEY                00000122  RG    06      LOCKID                    *********    X  03
DEFINE_SYMBOL             *********    X  03      LOCK_OPTS                 000000EA  R     03
DEVICE                    0000098E  R     03      LOG_FILE                  *********    X  03
DEVICE_OPTS               000009A2  R     03      MSG$_SUCCESS              *********    X  06
DISPLAY                   0000020E  R     03      MULT2                     00000009  R     06
DISPLAY_CRASH             *********    X  03      MULTIPLICATION            00000544  R     03
DISPLAY_DEVBYADDR         *********    X  03      NEG1                      00000035  R     06
DISPLAY_DEVICE            *********    X  03      NEGATE                    00000580  R     03
DISPLAY_HELP              *********    X  03      NXT_OPT                   00000220  R     03
DISPLAY_PFN               *********    X  03      OCTAL                     000005B6  R     03
DISPLAY_SPT               *********    X  03      OPEN_LOG                  *********    X  03
DISPLAY_SPT_RANGE         *********    X  03      OPEN_OUTPUT               *********    X  03
DIS_OPT                   0000022C  R     03      OPT$M_ALL               = 00000FFF
DIS_OPT_EXIT              00000248  R     03      OPT$M_ASB               = 00000040
DIV2                      0000001B  R     06      OPT$M_BAD               = 00000004
DIVISION                  0000054E  R     03      OPT$M_BDB               = 00000010
END_SET_RMS               00000192  R     06      OPT$M_BDBSUM            = 00000020
ESP                       *********    X  06      OPT$M_BLB               = 00008000
EVALUATE                  0000039E  R     03      OPT$M_BLBSUM            = 00010000
EVAL_CONSTANT             0000003E  R     06      OPT$M_CCB               = 00000080
EVAL_CONTENTS             00000087  R     06      OPT$M_CHAN              = 00000400
EVAL_DOT                  0000007A  R     06      OPT$M_COND              = 00000100
EVAL_EXP                  000003AE  R     03      OPT$M_ESP               = 00000004
EVAL_QUAL                 000003BE  R     03      OPT$M_FAB               = 00000400
EVAL_SETUP                00000048  R     06      OPT$M_FCB               = 00000200
EVAL_SYMBOL               00000062  R     06      OPT$M_FREE              = 00000001
```

```
OPT$M_FWA                       = 00100000          OPTIONS                         *********    X   03
OPT$M_GBD                       = 00020000          OUTPUT_FILE                     *********    X   03
OPT$M_GBDSUM                    = 00200000          PAGE_OPTION                     000008F8 R       03
OPT$M_GBH                       = 00040000          PAGE_OPTS                       000008E8 R       03
OPT$M_GLOBAL                    = 00000001          PAGE_RANGE                      0000091C R       03
OPT$M_HEADER                    = 00000080          PAGE_TABLE                      000008CC R       03
OPT$M_IDX                       = 00000008          PARENS                          00000594 R       03
OPT$M_IFB                       = 00000002          PDT_BYADDR                      *********    X   03
OPT$M_IMAGE                     = 00000001          PFN_DATA                        00000864 R       03
OPT$M_INST                      = 00000020          PFN_OPTION                      00000890 R       03
OPT$M_IRB                       = 00000004          PFN_OPTS                        00000886 R       03
OPT$M_IRP                       = 00000002          PIX_WIDTH                       *********    X   06
OPT$M_ISP                       = 00000001          POOL                            0000072E R       03
OPT$M_KSP                       = 00000002          POOL_RANGE                      000007C0 R       03
OPT$M_LCK                       = 00000200          PREV_BUFFER                     00000000 R       02
OPT$M_LENGTH                    = 00000010          PREV_COMMAND                    00000050 R       02
OPT$M_LRP                       = 00000020          PRINT_ANY_STACK                 *********    X   03
OPT$M_MODIFIED                  = 00000002          PRINT_HEADER                    *********    X   03
OPT$M_NAM                       = 00001000          PRINT_STACKS                    *********    X   03
OPT$M_NO                        = 00000001          PROCESS                         000005DA R       03
OPT$M_NONPAGED                  = 00000004          PROCESS1                        000005EE R       03
OPT$M_NOSKIP                    = 00000200          PROCESS_SUMMARY                 *********    X   03
OPT$M_P0                        = 00000001          PROC_INDEX                      *********    X   06
OPT$M_P0_PPT                    = 00000800          PROC_NAME                       *********    X   03
OPT$M_P1                        = 00000002          PROC_PIX                        000006B4 R       03
OPT$M_P1_PPT                    = 00001000          PROC_PPT                        000006C0 R       03
OPT$M_PAGED                     = 00000008          RADIX                           000005AA R       03
OPT$M_PCB                       = 00000010          READ_CMD                        000009F6 R       03
OPT$M_PHD                       = 00000020          READ_QUALS                      00000A0C R       03
OPT$M_PPT                       = 00000002          READ_SYMFILE                    *********    X   03
OPT$M_PPT_LEN                   = 00002000          RELOCATE_BASE                   *********    X   06
OPT$M_PPT_RNG                   = 00004000          REPEAT_COMMAND                  00000117 RG      06
OPT$M_PSL                       = 00000040          REPEAT_KEY                      *********    X   06
OPT$M_PST                       = 00000004          REQMEM                          *********    X   06
OPT$M_RAB                       = 00000800          RES                             00000102 R       03
OPT$M_RANGE                     = 00000008          RMS                             000006EC R       03
OPT$M_REGS                      = 00000008          RMS_DIS_OPT                     *********    X   06
OPT$M_RJB                       = 00400000          RMS_DIS_TMP                     *********    X   06
OPT$M_RLB                       = 00004000          RMS_DIS_TMP1                    *********    X   03
OPT$M_RMS                       = 00000080          RMS_IFI                         *********    X   06
OPT$M_RMSALL                    = FFFFFFFE          RMS_IFI_TMP                     *********    X   06
OPT$M_RMSD                      = 00000100          SAVE_COMMAND                    000000D3 RG      06
OPT$M_SELF                      = 00000001          SAVE_DUMP                       *********    X   03
OPT$M_SINGLEPFN                 = 00000010          SCS_CONNECT                     0000015E R       03
OPT$M_SRP                       = 00000200          SCS_PORTS                       00000178 R       03
OPT$M_SSP                       = 00000008          SCS_RDT                         00000192 R       03
OPT$M_SUMMARY                   = 00000040          SDA_KEY                         00000000 RG      04
OPT$M_SYSPROC                   = 00000040          SDA_STATE                       00000000 RG      03
OPT$M_SYSTEM                    = 00000004          SEARCH                          0000004E R       03
OPT$M_TIME                      = 00000080          SEARCH_MEMORY                   *********    X   03
OPT$M_TRC                       = 00080000          SEMI                          = 0000003B
OPT$M_TYPE                      = 00000100          SET                             000001B6 R       03
OPT$M_USP                       = 00000010          SETUP_DIS_OPT                   000001C8 R       06
OPT$M_WCB                       = 00000100          SETUP_RMS_TMP                   00000157 R       06
OPT$M_WHOLEPFN                  = 00000008          SET_LOG                         000001CE R       03
OPT$M_WSL                       = 00000001          SET_OUTPUT                      000001DE R       03
OPT$M_XAB                       = 00002000          SET_PROC                        00000700 R       03
```

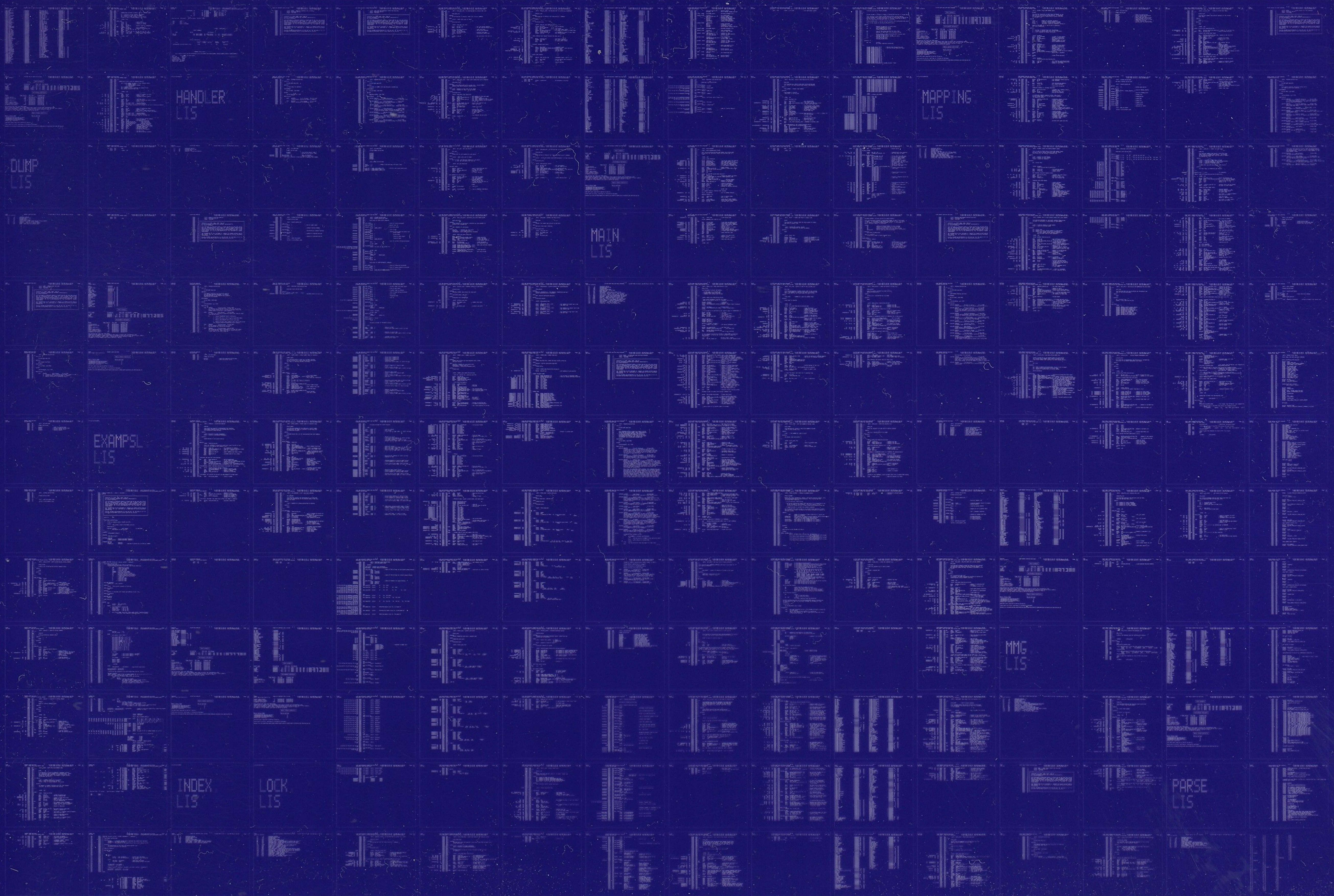
```
SET_PROCESS                     *********     X    03       TPA$_FAIL                     = FFFFFFFE
SET_RMS                         00000204 R         03       TPA$_FILESPEC                 = 000001EA
SHIFT2                          00000028 R         06       TPA$_HEX                      = 000001F5
SHOW                            00000088 R         03       TPA$_IDENT                    = 000001EC
SHOW_ALL_LOCKS                  *********     X    03       TPA$_KEYWORD                  = 00000100
SHOW_ALL_RES                    *********     X    03       TPA$_LAMBDA                   = 000001F6
SHOW_CLUSTER                    *********     X    03       TPA$_MAXKEY                   = 000000DC
SHOW_CONNECTIONS                *********     X    03       TPA$_OCTAL                    = 000001F4
SHOW_EXPR                       *********     X    03       TPA$_STRING                   = 000001F0
SHOW_ONE_LOCK                   *********     X    03       TPA$_SUBXPR                   = 000001F8
SHOW_ONE_RES                    *********     X    03       TPA$_SYMBOL                   = 000001F1
SHOW_POOL                       *********     X    03       TPA$_UIC                      = 000001EB
SHOW_POOL_RANGE                 *********     X    03       VALIDATE                        000004C2 R         03
SHOW_PORTS                      *********     X    03       VALIDATE_QUEUE                  *********     X    03
SHOW_PROCESS                    *********     X    03       VALIDATE_Q_OK                   000004F8 R         03
SHOW_RMS                        0000012C R         03       VAL_QUEUE                       000004C6 R         03
SHOW_RMS_OPT                    *********     X    03       VAL_QUEUE_OPTS                  000004D2 R         03
SHOW_RSPID                      *********     X    03       VAL_QUE_QUAL                    000004DC R         03
SHOW_SCS                        *********     X    03       VAL_SET_MAX                     *********     X    03
SHOW_SUMMARY                    00000942 R         03       VAXCLUSTER                      00000136 R         03
SHOW_SYMBOL                     *********     X    03
SMG$ADD_KEY_DEF                 *********     X    06
SMG$DEFINE_KEY                  *********     X    06
SMG$M_KEY_TERMINATE           = 00000002
SNARF                           000001FA R         03
STACK                           000007E6 R         03
STACK_RANGE                     0000083E R         03
START                           00000000 R         03
STORE                           00000050 R         06
STORE_IFI_TMP                   000001AE R         06
STORE_PROC_INDEX                000000BF R         06
STORE_RELOCATION                000000B4 R         06
STORE_STAR                      000001D4 R         06
STORE_TMP1_OPT                  00000169 R         06
STRUCTURE                       *********     X    03
SUB2                            00000012 R         06
SUBTRACTION                     0000051A R         03
SYM                             00000976 R         03
SYMBOL                          0000095A R         03
SYMBOL_DESC                     *********     X    03
SYMBOL_NAME                     *********     X    03
SYMBOL_OPTS                     00000982 R         03
SYMBOL_QUALS                    00000960 R         03
SYMBOL_VALUE                    *********     X    06
SYSREG                          000005CA R         03
TERM                            00000524 R         03
TPA$L_NUMBER                  = 0000001C
TPA$L_STRINGCNT               = 00000008
TPA$L_STRINGPTR               = 0000000C
TPA$L_TOKENCNT                = 00000010
TPA$_ALPHA                    = 000001EE
TPA$_ANY                      = 000001ED
TPA$_BLANK                    = 000001F2
TPA$_DECIMAL                  = 000001F3
TPA$_DIGIT                    = 000001EF
TPA$_EOS                      = 000001F7
TPA$_EXIT                     = FFFFFFFF
```

```
                                                     +----------------+
                                                     ! Psect synopsis !
                                                     +----------------+

PSECT name                         Allocation          PSECT No.  Attributes
----------                         ----------          ---------  ----------
.  ABS  .                          00000000  (     0.)  00 (  0.)  NOPIC   USR   CON   ABS   LCL NOSHR NOEXE NORD   NOWRT NOVEC BYTE
$ABS$                              00000000  (     0.)  01 (  1.)  NOPIC   USR   CON   ABS   LCL NOSHR   EXE   RD     WRT NOVEC BYTE
SDADATA                            0000005C  (    92.)  02 (  2.)  NOPIC   USR   CON   REL   LCL NOSHR NOEXE   RD     WRT NOVEC BYTE
_LIB$STATE$                        00000A5E  (  2654.)  03 (  3.)    PIC   USR   CON   REL   LCL   SHR   EXE   RD   NOWRT NOVEC BYTE
_LIB$KEY0$                         00000110  (   272.)  04 (  4.)    PIC   USR   CON   REL   LCL   SHR   EXE   RD   NOWRT NOVEC WORD
_LIB$KEY1$                         0000039E  (   926.)  05 (  5.)    PIC   USR   CON   REL   LCL   SHR   EXE   RD   NOWRT NOVEC WORD
PARSE                              00000201  (   513.)  06 (  6.)  NOPIC   USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC BYTE

                                                  +------------------------+
                                                  ! Performance indicators !
                                                  +------------------------+

Phase                        Page faults    CPU Time       Elapsed Time
-----                        -----------    --------       ------------
Initialization                        35    00:00:00.07    00:00:02.56
Command processing                   132    00:00:00.50    00:00:09.59
Pass 1                              1216    00:01:05.52    00:04:07.53
Symbol table sort                      0    00:00:00.61    00:00:02.04
Pass 2                               260    00:00:14.41    00:00:53.72
Symbol table output                   37    00:00:00.17    00:00:00.38
Psect synopsis output                  2    00:00:00.04    00:00:00.39
Cross-reference output                 0    00:00:00.00    00:00:00.00
Assembler run totals                1684    00:01:21.32    00:05:16.22

The working set limit was 3000 pages.
519055 bytes (1014 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 596 non-local and 3 local symbols.
1370 source lines were read in Pass 1, producing 69 object records in Pass 2.
32 pages of virtual memory were used to define 26 macros.

                                                 +--------------------------+
                                                 ! Macro library statistics !
                                                 +--------------------------+

Macro library name                                 Macros defined
------------------                                 --------------
_$255$DUA28:[SDA.OBJ]SDALIB.MLB;1                             4
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                                0
_$255$DUA28:[SYSLIB]STARLET.MLB;2                            10
TOTALS (all libraries)                                       14

753 GETS were required to define 14 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:PARSE/OBJ=OBJ$:PARSE MSRC$:PARSE/UPDATE=(ENH$:PARSE)+EXECML$/LIB+LIB$:SDALIB/LIB
```

POO
V04

0352 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



HANDLER
LIS

MAPPING
LIS

DUMP
LIS

MAIN
LIS

EXAMPSL
LIS

MMG
LIS

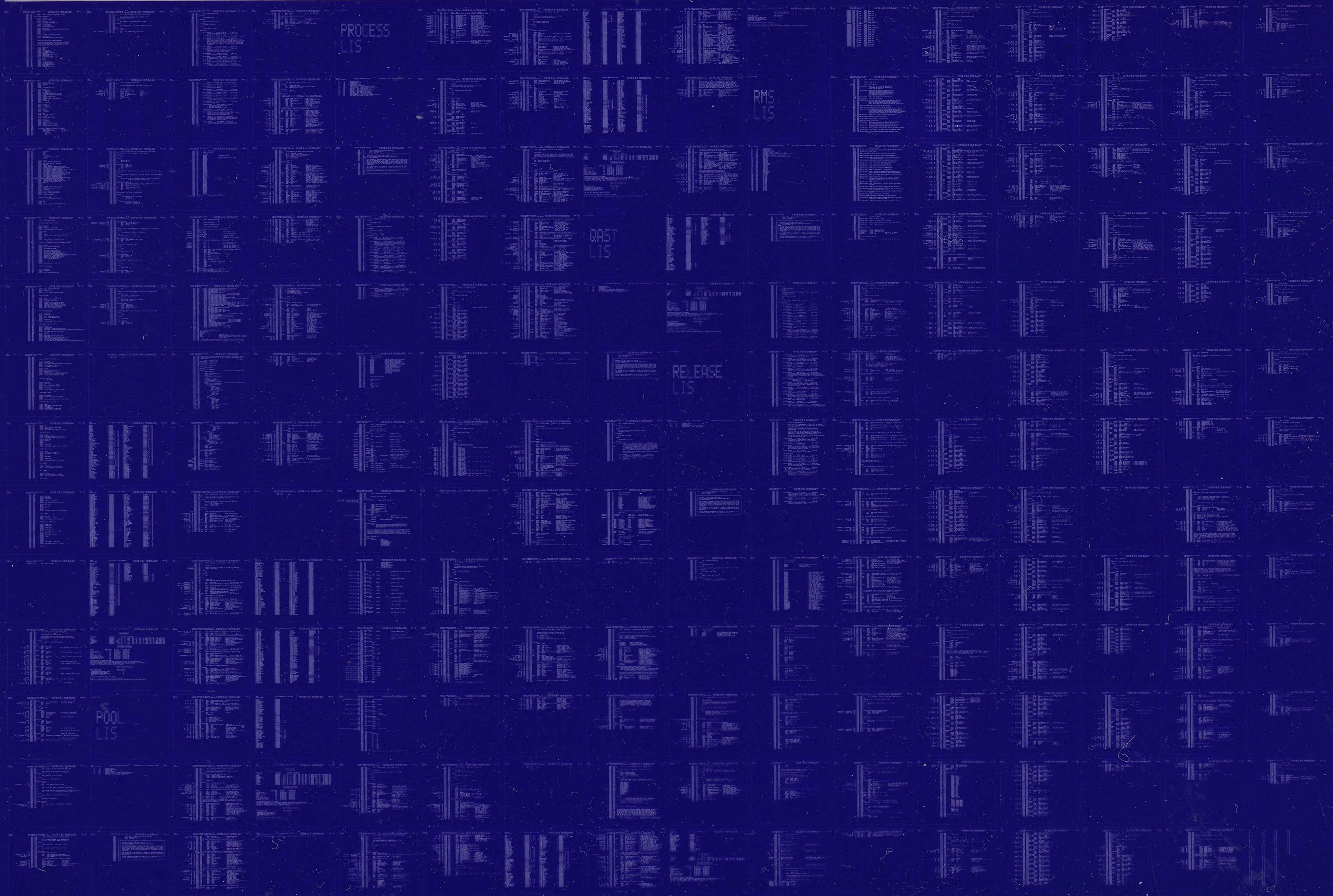
INDEX
LIS

LOCK
LIS

PARSE
LIS

0353 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



PROCESS
LIS

RMS
LIS

QAST
LIS

RELEASE
LIS

POOL
LIS