
SSSSSSSSSSS DDDDDDDDDDD AAAAAAAAA
SSSSSSSSSSS DDDDDDDDDDD AAAAAAAAA
SSSSSSSSSSS DDDDDDDDDDD 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
SSSSSSSSSSS DDDDDDDDDDD AAA AAA
SSSSSSSSSSS DDDDDDDDDDD AAA AAA
SSSSSSSSSSS DDDDDDDDDDD AAA AAA

FILEID**COMMANDS

```
  CCCCCCCC    OOOOOO    MM      MM  MM      MM    AAAAAA    NN      NN  DDDDDDDD      SSSSSSSS
  CCCCCCCC    OOOOOO    MM      MM  MM      MM    AAAAAA    NN      NN  DDDDDDDD      SSSSSSSS
CC           OO      OO  MMMM  MMMM  MMMM  MMMM  AA      AA  NN      NN  DD      DD  SS
CC           OO      OO  MMMM  MMMM  MMMM  MMMM  AA      AA  NN      NN  DD      DD  SS
CC           OO      OO  MM  MM  MM  MM  MM  MM  AA      AA  NNNN    NN  DD      DD  SS
CC           OO      OO  MM  MM  MM  MM  MM  MM  AA      AA  NNNN    NN  DD      DD  SS
CC           OO      OO  MM      MM  MM      MM  AA      AA  NN  NN  NN  DD      DD    SSSSSS
CC           OO      OO  MM      MM  MM      MM  AA      AA  NN  NN  NN  DD      DD    SSSSSS
CC           OO      OO  MM      MM  MM      MM  AAAAAAAAAA  NN    NNNN  DD      DD          SS
CC           OO      OO  MM      MM  MM      MM  AAAAAAAAAA  NN    NNNN  DD      DD          SS
CC           OO      OO  MM      MM  MM      MM  AA      AA  NN      NN  DD      DD          SS      ....
CC           OO      OO  MM      MM  MM      MM  AA      AA  NN      NN  DD      DD          SS      ....
  CCCCCCCC    OOOOOO    MM      MM  MM      MM  AA      AA  NN      NN  DDDDDDDD    SSSSSSSS        ....
  CCCCCCCC    OOOOOO    MM      MM  MM      MM  AA      AA  NN      NN  DDDDDDDD    SSSSSSSS        ....

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
```

Table of contents

```
0000     1          .TITLE  COMMANDS        PARSE AND EXECUTE SDA COMMANDS
0000     2          .SBTTL  COPYRIGHT NOTICE
0000     3          .IDENT  'V04-000'
0000     4 ;
0000     5 ;**********************************************************************
0000     6 ;*                                                                    *
0000     7 ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                           *
0000     8 ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.            *
0000     9 ;*  ALL RIGHTS RESERVED.                                              *
0000    10 ;*                                                                    *
0000    11 ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED *
0000    12 ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE *
0000    13 ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER *
0000    14 ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY *
0000    15 ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY *
0000    16 ;*  TRANSFERRED.                                                      *
0000    17 ;*                                                                    *
0000    18 ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE *
0000    19 ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT *
0000    20 ;*  CORPORATION.                                                      *
0000    21 ;*                                                                    *
0000    22 ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS *
0000    23 ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.           *
0000    24 ;*                                                                    *
0000    25 ;*                                                                    *
0000    26 ;**********************************************************************
0000    27 ;
```

```
0000     29          .SBTTL  PROGRAM DESCRIPTION
0000     30 ;++
0000     31 ;   FACILITY
0000     32 ;
0000     33 ;       SYSTEM DUMP ANALYZER
0000     34 ;
0000     35 ;   ABSTRACT
0000     36 ;
0000     37 ;       THIS MODULE ACCEPTS COMMANDS FROM THE PRIMARY
0000     38 ;       INPUT DEVICE AND PERFORMS THE SYNTAX PARSING.
0000     39 ;       THE CORRECT ROUTINE IS THEN CALLED TO EXECUTE
0000     40 ;       THE COMMAND.
0000     41 ;
0000     42 ;   ENVIRONMENT
0000     43 ;
0000     44 ;       NATIVE MODE, USER MODE
0000     45 ;
0000     46 ;   AUTHOR
0000     47 ;
0000     48 ;       TIM HALVORSEN, JULY 1978
0000     49 ;
0000     50 ;   MODIFIED BY
0000     51 ;
0000     52 ;       V03-006 PRB0305         Paul Beck       10-Jan-1984  16:23
0000     53 ;               Add support for EXAMINE/INSTRUCTION/NOSKIP
0000     54 ;
0000     55 ;       V03-005 WMC0001         Wayne Cardoza           13-Jul-1983
0000     56 ;               Add EVAL/CONDITION.
0000     57 ;
0000     58 ;       V03-004 JLV0258         Jake VanNoy             23-MAY-1983
0000     59 ;               Replace $GET with call to SMG$ routine for keypad input.
0000     60 ;
0000     61 ;       V03-003 TMK0001         Todd M. Katz            21-Mar-1983
0000     62 ;               Implement the logging of interactive sessions. When a logging
0000     63 ;               file has been defined, and the output is going to the terminal
0000     64 ;               and not to a listing file, then log each command line to the log
0000     65 ;               file before processing it.
0000     66 ;
0000     67 ;       V03-002 CWH1002         CW Hobbs        2-Mar-1983
0000     68 ;               Save pix width in a local so that we can extract the
0000     69 ;               process index width from either an internal pid or an
0000     70 ;               extended pid without even checking which we have.
0000     71 ;
0000     72 ;       V03-001 JLV0221         Jake VanNoy             21-JAN-1983
0000     73 ;               Add Examine/PSL Examime/CONDITION_VALUE and Examine/TIME.
0000     74 ;               Repeat "escape" refreshes command line.
0000     75 ;
0000     76 ;       V007    MTR0002         Mike Rhodes     27-Jul-1981
0000     77 ;               Change SDA help to not prompt.
0000     78 ;
0000     79 ;       V006    MTR0001         Mike Rhodes     19-Jun-1981
0000     80 ;               Change all CMPW's referencing an MSG$_ symbol to CMPL's.
0000     81 ;
0000     82 ;               Change default addressing mode to longword.
0000     83 ;
0000     84 ;               1.  Remove references to macro $SDAMSGDEF.
0000     85 ;
```

```
0000     86 ;                2.  Modify routine GET_COMMANDS reference to condition
0000     87 ;                handler entry point (from CMD_HANDLER to HANDLER) for
0000     88 ;                standardized error handling.
0000     89 ;
0000     90 ;                3.  Rewrite routine DISPLAY_HELP to use the librarian
0000     91 ;                routine LBR$OUTPUT_HELP.
0000     92 ;
0000     93 ;        V005    TMH0005         Tim Halvorsen   21-May-1981
0000     94 ;                Remove READ/RELOCATE modifications in previous update,
0000     95 ;                since the relocation must be done within ADD_SYMBOL
0000     96 ;                based on whether the symbol already exists or not (we
0000     97 ;                can't relocate the values for symbols which already exist,
0000     98 ;                such as structure definitions).
0000     99 ;
0000    100 ;        V004    TMH0004         Tim Halvorsen   20-May-1981
0000    101 ;                Upcase all commands.  Add '@' and SEARCH commands.
0000    102 ;                Add READ/RELOCATE qualifier.  Increase size of symbolize
0000    103 ;                string buffer on EXAMINE command.
0000    104 ;
0000    105 ;        V003    TMH0003         Tim Halvorsen   07-Feb-1981
0000    106 ;                Add EXAMINE/INSTRUCTION.
0000    107 ;
0000    108 ;        V002    TMH0002         Tim Halvorsen   19-Dec-1980
0000    109 ;                Fix current process context so that it remembers /SYSTEM
0000    110 ;                from one SHOW PROCESS to another.
0000    111 ;
0000    112 ;        V001    TMH0001         Tim Halvorsen   30-Sep-1980
0000    113 ;                Use general addressing to reference LIB$SIGNAL
0000    114 ;--
```

```
0000    116          .SBTTL  DECLARATIONS
0000    117 ;
0000    118 ;        SYMBOL DEFINTIONS
0000    119 ;
0000    120          $TPADEF                 ; TPARSE DEFINITIONS
0000    121          $PCBDEF                 ; PROCESS CONTROL BLOCK
0000    122          $PHDDEF                 ; PROCESS HEADER DEFINITIONS
0000    123          $OPTDEF                 ; DEFINE BITS IN OPTIONS WORD
0000    124          $SHRDEF                 ; SHARED ERROR MESSAGES
0000    125          $DMPDEF                 ; DUMP HEADER DEFINITIONS
0000    126          $EMBDEF <CR>            ; CRASH ERROR LOG ENTRY
0000    127          $FABDEF                 ; FILE ACCESS BLOCK
0000    128          $RABDEF                 ; RECORD ACCESS BLOCK
0000    129          $OBJDEF                 ; OBJECT MODULE DEFINITIONS
0000    130          $STSDEF                 ; MESSAGE FIELD DEFINITIONS
0000    131          $CHFDEF                 ; CONDITION HANDLER FIELD DEFINITIONS
0000    132          $SSDEF                  ; SYSTEM SERVICE CONDITION CODE DEF'S
0000    133          $HLPDEF                 ; HELP LIBRARY MASK DEFINITIONS
0000    134          $DVIDEF                 ; GETDVI DEFINITIONS
0000    135
```

```
                                0000   137          .SBTTL  STORAGE DEFINITIONS
                                0000   138 ;
                                0000   139 ;       STORAGE DEFINITIONS
                                0000   140 ;
                                0000   141
                             00000000  142          .PSECT  SDADATA,NOEXE,WRT
                                0000   143
                         0000000E  0000   144 HELP_FLAGS:     .LONG   HLP$M_PROCESS!- ;Library search & Prompt options
                                0004   145                             HLP$M_GROUP!-
                                0004   146                             HLP$M_SYSTEM
45 48 24 53 59 53 0000000C'010E0000'  0004   147 HELP_LIBRARY:    .ASCID  /SYS$HELP:SDA.HLB/
      42 4C 48 2E 41 44 53 3A 50 4C   0012
                                001C   148
                                001C   149 TPARSE_BLOCK:
                         00000008  001C   150          .LONG   TPA$K_COUNTO            ; LONGWORD IN BLOCK
                         00000002  0020   151          .LONG   TPA$M_ABBREV            ; ABBREVIATIONS ALLOWED
                         00000040  0024   152          .BLKL   TPA$K_COUNTO-1          ; REMAINDER OF BLOCK
                                0040   153
                                0040   154 ; STACK USED FOR EVALUATION OF EXPRESSIONS
                         0000010C' 0040   155 ESP::     .LONG   EXPR_STACK              ; STACK POINTER
                         0000010C  0044   156          .BLKL   50
                                010C   157 EXPR_STACK:                                 ; STARTING STACK POINTER
                                010C   158
                                010C   159 INPUT_RAB::
                         00000000  010C   160          .LONG   0                       ; ADDRESS OF CURRENT INPUT RAB
                                0110   161                                             ; INITIALLY 0
                                0110   162                                             ; (MAY BE CHANGED TO INDIRECT RAB)
                                0110   163
                                0110   164 OPTIONS::
                         00000114  0110   165          .BLKL   1                       ; OPTIONS FOR COMMAND
                                0114   166
                                0114   167 RELOCATE_BASE::
                         00000118  0114   168          .BLKL   1                       ; READ/RELOCATE BASE ADDRESS
                                0118   169
                                0118   170 PROC_NAME::
                         00000120  0118   171          .BLKQ   1                       ; PROCESS NAME
                                0120   172
                                0120   173 SYMBOL_NAME::
                         00000128  0120   174          .BLKQ   1                       ; SYMBOL NAME
                                0128   175
                                0128   176 FILE_DESC::
                         000001..  0128   177          .BLKQ   1                       ; FILE NAME DESCRIPTOR
                                0130   178
                                0130   179 PROC_INDEX::
                         FFFFFFFF  0130   180          .LONG   -1                      ; PROCESS INDEX
                                0134   181
                                0134   182 PIX_WIDTH::                                ; USEFUL WIDTH OF PROCESS INDEX, SO
                         00000138  0134   183          .BLKL   1                       ; WE CAN INTERCHANGE IPIDS AND EPIDS
                                0138   184
                                0138   185 PROC_PID::
                         0000013C  0138   186          .BLKL   1                       ; PROCESS PID
                                013C   187
                                013C   188 CURRENT_PID:
                         00000140  013C   189          .BLKL   1                       ; PID OF PROCESS RUNNING DURING CRASH
                                0140   190
                                0140   191 ADDRESS::
                         00000000  0140   192          .LONG   0                       ; CURRENT MEMORY ADDRESS
```

```
                     0144   193
                     0144   194 DATALEN:
            00000000 0144   195         .LONG   0                       ; LENGTH OF PREVIOUS DATA ITEM
                     0148   196
                     0148   197 SYMBOL_DESC::
            00000150 0148   198         .BLKQ   1                       ; SYMBOL NAME TO BE DEFINED
                     0150   199
                     0150   200 LOG_BUFFER:
            00000055 0150   201 LOG_BUFFER_LENGTH = 85
          20 3E 41 44 53 0150   202         .ASCII  'SDA> '
            000001A5 0155   203         .BLKB   LOG_BUFFER_LENGTH-5
                     01A5   204
                 00000000   205         .PSECT  COMMANDS,EXE,NOWRT,LONG
                     0000   206
                     0000   207 ;
                     0000   208 ; READ-ONLY STORAGE DEFINITIONS
                     0000   209 ;
                     0000   210
                     0000   211         .MACRO  NORMAL FIRST,LAST       ; MACRO TO DEFINE "NORMAL" INSTRUCTIONS
                     0000   212 .=NORMAL_INSTRS+FIRST
                     0000   213         .IF B   LAST                    ; IF LAST UNSPECIFIED, LOOP ONCE
                     0000   214         $$ = 1
                     0000   215         .IFF
                     0000   216         $$ = LAST+1-FIRST
                     0000   217         .ENDC
                     0000   218         .REPT   $$
                     0000   219         .BYTE   1
                     0000   220         .ENDR
                     0000   221         .ENDM
                     0000   222
                     0000   223 NORMAL_INSTRS:                       ; BYTEMASK OF REASONABLE INSTRUCTIONS
                     0000   224                                       ; (ITS ONLY A BYTEMASK CAUSE I DON'T
                     0000   225                                       ;  KNOW HOW TO CONSTRUCT THE BITMASK).
                     0000   226         .REPT   256/8
                     0000   227         .QUAD   0                       ; PRESET ALL TO "UNREASONABLE"
  00000000 00000000  0000   228         .ENDR
                     0100   229
                     0100   230         .SAVE                           ; SAVE CURRENT "."
                     0100   231         NORMAL  4,5                     ; RET THRU RSB
                     0006   232         NORMAL  12,27                   ; PROBER THRU BLEQU
                     001C   233         NORMAL  30,31                   ; BGEQU THRU BLSSU
                     0020   234         NORMAL  40,41                   ; MOVC3 THRU CMPC3
                     002A   235         NORMAL  44,45                   ; MOVC5 THRU CMPC5
                     002E   236         NORMAL  48,50                   ; BSBW THRU CVTWL
                     0033   237         NORMAL  58                      ; LOCC
                     003B   238         NORMAL  60                      ; MOVZWL
                     003D   239         NORMAL  120                     ; ASHL
                     0079   240         NORMAL  124,139                 ; CLRQ THRU BICB3
                     008C   241         NORMAL  144,145                 ; MOVB THRU CMPB
                     0092   242         NORMAL  147,156                 ; BITB THRU ROTL
                     009D   243         NORMAL  158,171                 ; MOVAB THRU BICW3
                     00AC   244         NORMAL  176,183                 ; MOVW THRU DECW
                     00B8   245         NORMAL  186,187                 ; PUSHR THRU POPR
                     00BC   246         NORMAL  192,206                 ; ADDL2 THRU MNEGL
                     00CF   247         NORMAL  208,215                 ; MOVL THRU DECL
                     00D8   248         NORMAL  218,233                 ; MTPR THRU BLBC
                     00EA   249         NORMAL  236,240                 ; CMPV THRU INSV
```

```
                                                E 4
                00F1   250          NORMAL  242,245                    ; AOBLSS THRU SOBGTR
                00F6   251          NORMAL  250,251                    ; CALLG THRU CALLS
                00FC   252          NORMAL  255                        ; BUGCHECK
            00000100   253          .RESTORE
                0100   254
                0100   255          .DEFAULT DISPLACEMENT,LONG
```

```
                                0100   257          .SBTTL  GET_COMMANDS -- ACCEPT AND EXECUTE COMMANDS
                                0100   258 ;---
                                0100   259 ;
                                0100   260 ;       GET_COMMANDS
                                0100   261 ;
                                0100   262 ;       THIS ROUTINE ACCEPTS THE NEXT COMMAND FROM THE PRIMARY
                                0100   263 ;       INPUT FILE, PARSES THE COMMAND AND EXECUTES IT.
                                0100   264 ;
                                0100   265 ;   INPUTS:
                                0100   266 ;
                                0100   267 ;       NONE
                                0100   268 ;
                                0100   269 ;   OUTPUTS:
                                0100   270 ;
                                0100   271 ;       R0 = TRUE IF COMMAND EXECUTED, FALSE IF EMPTY LINE
                                0100   272 ;
                                0100   273 ;---
                                0100   274          .ENABL  LSB
                                0100   275
                                0100   276 PROMPT:
                    00000005    0100   277 PROMPT_LEN = 5
 20 3E 41 44 53 0A 00'          0100   278          .ASCIC  <10>'SDA> '
                          06    0100
                                0107   279
                                0107   280 GET_COMMANDS::
                        000C    0107   281          .WORD   ^M<R2,R3>
                                0109   282
    6D  00000000'EF     DE      0109   283          MOVAL   HANDLER,(FP)            ; SETUP CONDITION HANDLER
    53  0000001C'EF     9E      0110   284          MOVAB   TPARSE_BLOCK,R3         ; SET ADDRESS OF TPARSE BLOCK
                                0117   285 ;        WARNING -- THE HANDLER KNOWS ABOUT THE FOLLOWING 2 LINES
        00000110'EF     DD      0117   286          PUSHL   OPTIONS                 ; SAVE OPTIONS LONGWORD
    7E  00000000'EF     7D      011D   287          MOVQ    SUB_HEADING,-(SP)       ; SAVE CURRENT HEADING
                                0124   288
        00000000'EF     D4      0124   289          CLRL    LINE_COUNT              ; 1ST PAGE EJECT W/O PROMPT
        0000010C'EF     D5      012A   290          TSTL    INPUT_RAB               ; indirect?
                          59    12      0130   291          BNEQ    20$
        00000000'EF     16      0132   292          JSB     GET_INPUT               ; Get a line of input
    50  00000000'8F     D1      0138   293          CMPL    #SMG$_EOF,R0            ; CHECK IF END OF FILE
                          12    12      013F   294          BNEQ    10$             ; BRANCH IF NOT
                                0141   295          SIGNAL  0,EOF                   ; SIGNAL END OF FILE
                                0153   296 10$:
                                0153   297          SIGNAL                          ; CHECK FOR OTHER ERRORS
                          7D    11      015F   298          BRB     45$
                                0161   299 ;
                                0161   300 ;       IF THE ESCAPE KEY WAS TYPED, AND LINE EDITING IS OFF,
                                0161   301 ;       REPEAT THE LAST COMMAND.
                                0161   302 ;
1E 00000000'EF  00000000'8F     E1      0161   303          BBC     #DEV$V_TRM,DVI_DEVCHAR,20$ ; BR IF NOT TERMINAL
                00000000'8F     E0      016D   304          BBS     #TT2$V_EDITING,-
              12 00000000'EF            0173   305                  DVI_DEVDEPND2,20$  ; BR IF EDITING
                  1B    0C A2   B1      0179   306          CMPW    RAB$L_STV(R2),#^X1B ; IS TERMINATOR ESCAPE KEY?
                          5F    12      017D   307          BNEQ    45$             ; BRANCH IF NOT
            00000000'EF   63    FA      017F   308          CALLG   (R3),REPEAT_COMMAND ; RESTORE PREV. COMMAND
                        0AAE    30      0186   309          BSBW    ECHO            ; ECHO IF SCOPE
                          6A    11      0189   310          BRB     50$             ; AND EXECUTE IT
                                0188   311          ;
                                0188   312          ; Use indirect file
```

```
                                018B   313          ;
                                018B   314 20$:
   52   0000010C'EF    D0      018B   315          MOVL    INPUT_RAB,R2                ; Set rab address
24 A2   00000000'EF    9E      0192   316          MOVAB   INPUT_BUFFER,RAB$L_UBF(R2)
   20 A2    0000'8F    B0      019A   317          MOVW    #INPUT_BUF_LEN,RAB$W_USZ(R2)
                               01A0   318          $GET    (R2)
00000000'EF    22 A2    B0     01A9   319          MOVW    RAB$W_RSZ(R2),INPUT_LEN ; Set length
   50   00000000'8F    D1      01B1   320          CMPL    #RMS$_EOF,R0                ; CHECK IF END OF FILE
                12    12      01B8   321          BNEQ    30$                         ; BRANCH IF NOT
                               01BA   322          SIGNAL  0,EOF                       ; SIGNAL END OF FILE
                               01CC   323 30$:
                               01CC   324          SIGNAL  RMS,(R2)                    ; CHECK FOR OTHER ERRORS
                               01DE   325 ;
                               01DE   326 ;        PARSE THE COMMAND
                               01DE   327 ;
0C A3   00000000'EF    9E      01DE   328 45$:     MOVAB   INPUT_BUFFER,TPA$L_STRINGPTR(R3)
08 A3   00000000'EF    3C      01E6   329          MOVZWL  INPUT_LEN,TPA$L_STRINGCNT(R3)
                05    12      01EE   330          BNEQ    50$                         ; BRANCH IF NOT EMPTY
                50    D4      01F0   331          CLRL    R0                          ; INDICATE EMPTY LINE
              00CF    31      01F2   332          BRW     90$                         ; EXIT
                               01F5   333 50$:
        00000000'EF    D5      01F5   334          TSTL    OUTPUT_FILE                 ; SKIP LOGGING IF RMS IS OUTPUTTING
                4A    12      01FB   335          BNEQ    55$                         ; TO LISTING FILE OR LOGGING FILE
        00000000'EF    D5      01FD   336          TSTL    LOG_FILE                    ; HAS NOT BEEN DEFINED
                42    13      0203   337          BEQL    55$
                               0205   338
            007C 8F    BB      0205   339          PUSHR   #^M<R2,R3,R4,R5,R6>         ; SAVE SOME REGISTERS
   56   00000000'EF    9E      0209   340          MOVAB   LOGRAB,R6                   ; LOGGING FILE RAB
28 A6   00000150'EF    9E      0210   341          MOVAB   LOG_BUFFER,RAB$L_RBF(R6)    ; COMMAND INPUT LINE LOGGING BUFFER
                05    A1      0218   342          ADDW3   #PROMPT_LEN,-               ; LENGTH OF LOGGING I/O IS SIZE OF
              08 A3            021A   343                  TPA$L_STRINGCNT(R3),-       ; PROMPT PLUS SIZE OF COMMAND INPUT LINE
              22 A6            021C   344                  RAB$W_RSZ(R6)
              08 A3    28      021E   345          MOVC3   TPA$L_STRINGCNT(R3),-       ; APPEND THE COMMAND INPUT LINE AFTER
              0C B3            0221   346                  @TPA$L_STRINGPTR(R3),-      ; THE PROMPT IN THE LOGGING BUFFER
        00000155'EF            0223   347                  LOG_BUFFER+PROMPT_LEN
                               0228   348          $PUT    (R6)                        ; LOG THE COMMAND INPUT LINE
                               0231   349          SIGNAL  RMS,(R6)                    ; SIGNAL ANY ERRORS
            007C 8F    BA      0243   350          POPR    #^M<R2,R3,R4,R5,R6>         ; RESTORE SAVE REGISTERS
                               0247   351
              08 A3    9F      0247   352 55$:     PUSHAB  TPA$L_STRINGCNT(R3)         ; ADDRESS OF INPUT STRING DESCRIPTOR
                 6E    DD      024A   353          PUSHL   (SP)                        ; OVERWRITE STRING WITH RESULT
     00000000'GF      02      FB     024C   354          CALLS   #2,G^STR$UPCASE             ; UPCASE COMMAND LINE
          00000110'EF    D4      0253   355          CLRL    OPTIONS                     ; INITIALIZE OPTIONS LONGWORD
          00000114'EF    D4      0259   356          CLRL    RELOCATE_BASE               ; INITIALIZE RELOCATION BASE
00000040'EF    0000010C'EF    DE      025F   357          MOVAL   EXPR_STACK,ESP              ; INITIALIZE EXPRESSION STACK
            7E    08 A3    7D      026A   358          MOVQ    TPA$L_STRINGCNT(R3),-(SP)   ; SAVE DESCRIPTOR OF COMMAND
          00000000'EF    DF      026E   359          PUSHAL  L^SDA_KEY                   ; KEY TABLE ADDRESS
          00000000'EF    DF      0274   360          PUSHAL  L^SDA_STATE                 ; STATE TABLE ADDRESS
                53    DD      027A   361          PUSHL   R3                          ; TPARSE BLOCK
     00000000'GF      03      FB     027C   362          CALLS   #3,G^LIB$TPARSE             ; PARSE AND EXECUTE THE COMMAND
          00000000'EF    D4      0283   363          CLRL    HEADING_ROUTINE             ; CLEAR HEADING ROUTINE ADDRESS
            08 A3    8E      7D     0289   364          MOVQ    (SP)+,TPA$L_STRINGCNT(R3)   ; RESTORE COMMAND DESCRIPTOR
                2D 50    E8      028D   365          BLBS    R0,80$                      ; BRANCH IF NO ERROR
                               0290   366 ;
                               0290   367 ;        SIGNAL SYNTAX ERROR FROM PARSE
                               0290   368 ;
   51   50   10   10    EF     0290   369          EXTZV   #16,#16,R0,R1               ; CHECK FACILITY CODE
```

```
                 0E   12  0295   370          BNEQ    60$                     ; BRANCH IF NOT SDA ERROR MSG
                                 0297   371          SIGNAL                  ; SIGNAL ERROR CONDITION
                 18   11  02A3   372          BRB     80$
                                 02A5   373 60$:
50  000310F8 8F   DO  02A5   374          MOVL    #SHR$_SYNTAX!<3@16>,R0  ; CLI-W-SYNTAX MESSAGE
          10 A3   9F  02AC   375          PUSHAB  TPA$L_TOKENCNT(R3)      ; DESCRIPTOR OF BAD TOKEN
                      02AF   376          SIGNAL  1                       ; SIGNAL WITH 1 FAO ARG.
                      02BD   377 80$:
                      02BD   378          STATUS  SUCCESS                 ; COMMAND EXECUTED
                      02C4   379 90$:
00000000'EF   8E   7D  02C4   380          MOVQ    (SP)+,SUB_HEADING       ; RESTORE SUB_HEADING
     00000110'EF 8ED0  02CB   381          POPL    OPTIONS                 ; RESTORE OPTIONS WORD
                 04   02D2   382          RET
                      02D3   383
                      02D3   384          .DSABL  LSB
```

```
                                02D3    386          .SBTTL  INDIRECT_COMMAND --INDIRECT FILE PROCESSING
                                02D3    387 ;+
                                02D3    388 ;       INDIRECT_COMMAND
                                02D3    389 ;
                                02D3    390 ; INPUTS:
                                02D3    391 ;
                                02D3    392 ;       AP = ADDRESS OF TPARSE CONTROL BLOCK
                                02D3    393 ;
                                02D3    394 ; OUTPUTS:
                                02D3    395 ;
                                02D3    396 ;       ACTION ROUTINE TO OPEN AN INDIRECT COMMAND FILE
                                02D3    397 ;       AND PROCESS THE COMMANDS.
                                02D3    398 ;-
                        003C    02D3    399          .ENTRY  INDIRECT_COMMAND, ^M<R2, R3, R4, R5>
                                02D5    400
    53  00000000'EF   9E    02D5    401          MOVAB   INDRAB, R3                  ; R3 = ADDRESS OF RAB
        52    3C A3   D0    02DC    402          MOVL    RAB$L_FAB(R3),R2            ; R2 = ADDRESS OF FAB
    53  0000010C'EF   D1    02E0    403          CMPL    INPUT_RAB, R3               ; ONLY ONE LEVEL OF INDIRECTION
                37    13    02E7    404          BEQL    10$                         ; BR IF RECURSIVE AND EXIT
                                02E9    405
              10 AC   33    02E9    406          CVTWB   TPA$L_TOKENCNT(AP),-
              34 A2                 02EC    407                  FAB$B_FNS(R2)       ; SET FILE NAME LENGTH
              14 AC   D0    02EE    408          MOVL    TPA$L_TOKENPTR(AP),-
              2C A2                 02F1    409                  FAB$L_FNA(R2)       ; SET FILE NAME ADDRESS
                                02F3    410
                                02F3    411          $OPEN   (R2)                    ; ATTEMPT TO OPEN THE FILE
              24 50   E8    02FC    412          BLBS    R0, 20$                     ; BRANCH IF OK
              2C A2   DD    02FF    413          PUSHL   FAB$L_FNA(R2)               ; CREATE DESCRIPTOR OF FILE NAME
        7E    34 A2   9A    0302    414          MOVZBL  FAB$B_FNS(R2), -(SP)
           7E    50   7D    0306    415          MOVQ    R0, -(SP)                   ; RMS ERROR CODES
                 00   F0    0309    416          INSV    #STS$K_WARNING,-            ; CHANGE SEVERITY TO WARNING
                 00         030B    417                  #STS$V_SEVERITY,-
                 03         030C    418                  #STS$S_SEVERITY,-
                 6E         030D    419                  (SP)
              08 AE   9F    030E    420          PUSHAB  8(SP)                       ; ADDRESS OF FILE NAME DESCRIPTOR
                 01   DD    0311    421          PUSHL   #1                          ; NUMBER OF FAO ARGUMENTS
        00031098   8F   DD    0313    422          PUSHL   #SHR$_OPENIN!<3@16>         ; "ERROR OPENING INPUT FILE !AS"
    00000000'GF    05   FB    0319    423          CALLS   #5, G^LIB$SIGNAL            ; OUTPUT ERROR MESSAGE
              00A4   31    0320    424 10$:     BRW     50$                         ; EXIT WITH SUCCESS
                                0323    425
                                0323    426 20$:     $CONNECT (R3)                   ; CONNECT FOR RECORD ACCESS
                                032C    427          SIGNAL  RMS, (R3)               ; REPORT ANY ERRORS
                                033E    428
    00000000'EF    6C   FA    033E    429          CALLG   (AP),REPEAT_COMMAND         ; GET DESCRIPTOR OF @ COMMAND
           5E   08 AC   C2    0345    430          SUBL    TPA$L_STRINGCNT(AP),SP      ; ALLOCATE BUFFER TO HOLD @ COMMAND
                 0C   BB    0349    431          PUSHR   #^M<R2,R3>
08 AE   0C BC   08 AC   28    034B    432          MOVC    TPA$L_STRINGCNT(AP),@TPA$L_STRINGPTR(AP),8(SP) ; PUSH ONTO STACK
                 0C   BA    0352    433          POPR    #^M<R2,R3>
              08 AC   DD    0354    434          PUSHL   TPA$L_STRINGCNT(AP)         ; SAVE LENGTH OF @ COMMAND STRING
        0000010C'EF   DD    0357    435          PUSHL   INPUT_RAB                   ; SAVE RAB ACROSS PROCEDURE
    0000010C'EF   53    D0    035D    436          MOVL    R3, INPUT_RAB               ; SETUP INDIRECT FILES RAB AS INPUT
                                0364    437 30$:
        00000000'EF   D4    0364    438          CLRL    LINE_COUNT                  ; AVOID END OF PAGE PROMPTS
           FD98 CF    00    FB    036A    439          CALLS   #0, GET_COMMANDS            ; ACCEPT AND EXECUTE COMMANDS
    00000000'8F    50   D1    036F    440          CMPL    R0, #MSG$_EOF               ; CHECK IF END OF FILE
                 1E   13    0376    441          BEQL    40$                         ; BRANCH IF SO
    00000000'8F    50   D1    0378    442          CMPL    R0, #MSG$_EXITCMD           ; DID WE JUST EXIT A LEVEL?
```

```
                    E3   12   037F   443          BNEQ    30$                          ; BRANCH IF NOT
00000000'EF         EF   7C   0381   444          CLRQ    SUB_HEADING                  ; CLEAR CURRENT HEADING
00000000'EF         EF   D4   0387   445          CLRL    HEADING_ROUTINE              ; CLEAR HEADING ROUTINE ADDRESS
                         038D   446          SKIP    PAGE                         ; ERASE PREVIOUS JUNK
                    8D   11   0394   447          BRB     20$
                         0396   448 40$:
0000010C'EF         EF 8ED0   0396   449          POPL    INPUT_RAB                    ; RESTORE RAB ADDRESS
              08    AC 8ED0   039D   450          POPL    TPA$L_STRINGCNT(AP)          ; RESTORE LENGTH OF @ STRING
           0C AC    5E   D0   03A1   451          MOVL    SP,TPA$L_STRINGPTR(AP)       ; SET ADDRESS OF COPY OF STRING
00000000'EF         EF   6C   FA   03A5   452          CALLG   (AP),SAVE_COMMAND            ; SAVE AS PREVIOUS COMMAND
                         03AC   453          $CLOSE  FAB=(R2)                     ; CLOSE THE COMMAND FILE
                         03B5   454          SIGNAL  RMS,(R2)                     ; REPORT ANY ERRORS
                         03C7   455 50$:
                    50   01   D0   03C7   456          MOVL    #1, R0                       ; RETURN SUCCESS
                         04   03CA   457          RET
```

```
                        .SBTTL  EXIT_COMMAND -- PROCESS EXIT COMMAND
                ;---
                ;
                ;       EXIT_COMMAND
                ;
                ;       EXIT COMMAND ACTION ROUTINE
                ;
                ;   OUTPUTS:
                ;
                ;       END OF FILE IS SIGNALED
                ;
                ;---

                EXIT_COMMAND::
0000            .WORD   0

                SIGNAL  0,EOF                   ; SIGNAL END OF FILE
```

```
                        03DF    477          .SBTTL  DEFINE_SYMBOL -- DEFINE LOCAL SYMBOL
                        03DF    478 ;---
                        03DF    479 ;
                        03DF    480 ;       DEFINE_SYMBOL
                        03DF    481 ;
                        03DF    482 ;       THIS ROUTINE ADDS A SYMBOL TO THE LOCAL SYMBOL
                        03DF    483 ;       TABLE WITH AN ASSOCIATED VALUE.
                        03DF    484 ;
                        03DF    485 ;   INPUTS:
                        03DF    486 ;
                        03DF    487 ;       AP = ADDRESS OF TPARSE PARAMETER BLOCK
                        03DF    488 ;       SYMBOL_DESC = DESCRIPTOR OF SYMBOL
                        03DF    489 ;       TPA$L_NUMBER(AP) = VALUE TO BE ASSIGNED TO SYMBOL
                        03DF    490 ;
                        03DF    491 ;---
                        03DF    492
                        03DF    493 DEFINE_SYMBOL::
                0000    03DF    494          .WORD   0
                        03E1    495
            1C AC   DD  03E1    496          PUSHL   TPA$L_NUMBER(AP)                ; VALUE OF SYMBOL
7E  00000148'EF     7D  03E4    497          MOVQ    SYMBOL_DESC,-(SP)               ; SYMBOL DESCRIPTOR
00000000'EF   03    FB  03EB    498          CALLS   #3,ADD_SYMBOL                   ; ADD TO SYMBOL TABLE
                    04  03F2    499          RET
```

```
                                03F3    501          .SBTTL  SHOW_SYMBOL -- DISPLAY SYMBOL VALUE
                                03F3    502 ;---
                                03F3    503 ;
                                03F3    504 ;       SHOW_SYMBOL
                                03F3    505 ;
                                03F3    506 ;       THIS ROUTINE DISPLAYS THE VALUE OF A SPECIFIED SYMBOL
                                03F3    507 ;       AND THE CONTENTS OF THAT MEMORY LOCATION IF POSSIBLE.
                                03F3    508 ;
                                03F3    509 ;   INPUTS:
                                03F3    510 ;
                                03F3    511 ;       SYMBOL_NAME = DESCRIPTOR OF SYMBOL
                                03F3    512 ;
                                03F3    513 ;   OUTPUTS:
                                03F3    514 ;
                                03F3    515 ;       NONE
                                03F3    516 ;
                                03F3    517 ;---
                                03F3    518
                                03F3    519          .ENABL  LSB
                                03F3    520
                                03F3    521 SHOW_SYMBOL::
                           0004 03F3    522          .WORD   ^M<R2>
                                03F5    523
      52   00000120'EF    9E   03F5    524          MOVAB   SYMBOL_NAME,R2          ; ADDRESS OF SYMBOL NAME DESC.
00000FFF 8F   00000110'EF    D1   03FC    525          CMPL    OPTIONS,#OPT$M_ALL      ; CHECK IF WANTS ALL SYMBOLS
                        6B   13   0407    526          BEQL    70$                     ; BRANCH IF SO
                        52   DD   0409    527          PUSHL   R2                      ; DESCRIPTOR
           00000000'EF  01   FB   040B    528          CALLS   #1,SYMBOL_VALUE         ; GET SYMBOL VALUE
                     4A 50   E9   0412    529          BLBC    R0,50$                  ; IF NOT FOUND
           C0000000 8F   51   D1   0415    530          CMPL    R1,#^XC0000000          ; CHECK IF INTERNAL SYMBOL
                        41   1E   041C    531          BGEQU   50$                     ; DO NOT SHOW INTERNAL SYMBOLS
           00000140'EF  51   D0   041E    532          MOVL    R1,ADDRESS              ; SAVE SYMBOL VALUE
                             0425    533          TRYMEM  (R1)                    ; GET CONTENTS OF LOCATION
                     18 50   E9   042E    534          BLBC    R0,20$                  ; IF NOT AN ADDRESS
                        51   DD   0431    535          PUSHL   R1                      ; CONTENTS OF LOCATION
                00000140'EF  DD   0433    536          PUSHL   ADDRESS                 ; SYMBOL VALUE
                        52   DD   0439    537          PUSHL   R2                      ; DESCRIPTOR ADDRESS
                             043B    538          PRINT   3,<!AS = !XL :  !XL>
                        04   0448    539          RET
                             0449    540 20$:
                00000140'EF  DD   0449    541          PUSHL   ADDRESS                 ; SYMBOL VALUE
                        52   DD   044F    542          PUSHL   R2                      ; DESCRIPTOR ADDRESS
                             0451    543          PRINT   2,<!AS = !XL>
                        04   045E    544          RET
                             045F    545 50$:
                        52   DD   045F    546          PUSHL   R2                      ; DESCRIPTOR ADDRESS
                             0461    547          SIGNAL  1,BADSYM                ; UNKNOWN SYMBOL
                        04   0473    548          RET
                             0474    549 70$:
00000000'EF             62   FA   0474    550          CALLG   (?2),PRINT_SYMBOLS      ; PRINT ALL SYMBOLS
                        04   047B    551          RET
                             047C    552
                             047C    553          .DSABL  LSB
```

```
                                .SBTTL  SHOW_EXPR -- SHOW RESULT OF EVALUATE COMMAND
                        ;---
                        ;
                        ;       SHOW_EXPR
                        ;
                        ;       DISPLAY THE RESULT OF AN EVALUATE COMMAND.
                        ;
                        ;   INPUTS:
                        ;
                        ;       TPA$L_NUMBER(AP) = RESULT OF EXPRESSION
                        ;
                        ;   OUTPUTS:
                        ;
                        ;       RESULT IS DISPLAYED.
                        ;
                        ;---

                        SHOW_EXPR::
                  0000  047C    .WORD   0
                        047E    .ENABL  LSB
11 00000110'EF    08    E0    047E    BBS     #OPT$V_COND,OPTIONS,10$ ; IS IT /COND
                  1C AC   DD    0486    PUSHL   TPA$L_NUMBER(AP)        ; GET RESULT OF EXPRESSION
                          0489    PRINT   1,<Hex = !XL   Decimal = !-!SL>
                     04   0496    RET
                          0497
                          0497  10$:    ALLOC   110,R2                  ; GET A BUFFER
                          04A9    $GETMSG_S -                             ; GET THE MESSAGE
                          04A9            MSGID = TPA$L_NUMBER(AP),-
                          04A9            MSGLEN  = (R2),-
                          04A9            BUFADR = (R2)
                  10 50   E9    04BB    BLBC    R0,20$
                     52   DD    04BE    PUSHL   R2
                          04C0    PRINT   1,<!AS>
                     04   04CD    RET
                          04CE
                     50   DD    04CE    20$:    PUSHL   R0                      ; ERROR
00000000'GF          01    FB   04D0    CALLS   #1,G^LIB$SIGNAL
                     04   04D7    RET
                          04D8    .DSABL  LSB
```

```
                                04D8   595          .SBTTL  EXAM_MEMORY -- EXAMINE MEMORY LOCATIONS
                                04D8   596 ;---
                                04D8   597 ;
                                04D8   598 ;       EXAM_MEMORY
                                04D8   599 ;
                                04D8   600 ;       DISPLAY THE CONTENTS OF A SPECIFIED RANGE OF MEMORY
                                04D8   601 ;
                                04D8   602 ;   INPUTS:
                                04D8   603 ;
                                04D8   604 ;       EXPRESSION STACK CONTAINS ARGUMENTS OF RANGE
                                04D8   605 ;
                                04D8   606 ;   OUTPUTS:
                                04D8   607 ;
                                04D8   608 ;       NONE
                                04D8   609 ;
                                04D8   610 ;---
                                04D8   611
                                04D8   612          .ENABL  LSB
                                04D8   613
                                04D8   614 EXAM_MEMORY::
                        003C    04D8   615          .WORD   ^M<R2,R3,R4,R5>
                                04DA   616
     52    00000110'EF    D0    04DA   617          MOVL    OPTIONS,R2
50   52    000003F8 8F    CB    04E1   618          BICL3   #OPT$M_RANGE!-
                                04E9   619                  OPT$M_LENGTH!-
                                04E9   620                  OPT$M_INST!-
                                04E9   621                  OPT$M_NOSKIP!-
                                04E9   622                  OPT$M_PSL!-
                                04E9   623                  OPT$M_COND!-
                                04E9   624                  OPT$M_TIME,R2,R0        ; CHECK IF ANY OPTIONS SPECIFIED
           22    13    04E9   625          BEQL    40$                             ; BRANCH IF NOT
     07 52       00    E1    04EB   626          BBC     #OPT$V_P0,R2,10$        ; CHECK IF /P0 SPECIFIED
00000000'EF      00    FB    04EF   627          CALLS   #0,SHOW_P0              ; SHOW P0 SPACE
     07 52       01    E1    04F6   628 10$:     BBC     #OPT$V_P1,R2,20$        ; CHECK IF /P1 SPECIFIED
00000000'EF      00    FB    04FA   629          CALLS   #0,SHOW_P1              ; SHOW P1 SPACE
     07 52       02    E1    0501   630 20$:     BBC     #OPT$V_SYSTEM,R2,30$    ; CHECK IF /SYSTEM SPECIFIED
00000000'EF      00    FB    0505   631          CALLS   #0,SHOW_SYSTEM          ; SHOW SYSTEM SPACE
                   04    050C   632 30$:     RET
                         050D   633
                         050D   634 40$:
53   00000040'EF    D0    050D   635          MOVL    ESP,R3                  ; EXPRESSION STACK POINTER
     22 52    03    E0    0514   636          BBS     #OPT$V_RANGE,R2,50$     ; BRANCH IF RANGE SPECIFIED
     29 52    04    E0    0518   637          BBS     #OPT$V_LENGTH,R2,55$    ; BRANCH IF LENGTH SPECIFIED
50   0000010C'EF    9E    051C   638          MOVAB   EXPR_STACK,R0
        50       53    D1    0523   639          CMPL    R3,R0                   ; CHECK IF STACK EMPTY
                 5A    13    0526   640          BEQLU   60$                     ; IF NO ARGS, USE NEXT LONGWORD
C0000000 8F      63    D1    0528   641          CMPL    (R3),#^XC0000000        ; CHECK IF INTERNAL REGISTER
                 5F    1E    052F   642          BGEQU   70$                     ; DO NOT SET ADDRESS IF SO
00000140'EF      63    D0    0531   643          MOVL    (R3),ADDRESS            ; SET NEW "CURRENT" ADDRESS
                 56    11    0538   644          BRB     70$
                         053A   645 50$:
                 7E    D4    053A   646          CLRL    -(SP)                   ; DISABLE PAGING
 7E   63    04   A3    C3    053C   647          SUBL3   4(R3),(R3),-(SP)        ; LENGTH - 1 OF RANGE
                 6E    D6    0541   648          INCL    (SP)                    ; LENGTH TO DUMP
                 04    11    0543   649          BRB     58$
                         0545   650 55$:
                 7E    D4    0545   651          CLRL    -(SP)                   ; DISABLE PAGING
```

```
                     63   DD  0547   652          PUSHL   (R3)                          ; LENGTH TO DUMP
                          0549   653 58$:
                   04 A3   DD  0549   654          PUSHL   4(R3)                        ; STARTING ADDRESS
        00000140'EF   6E   D0  054C   655          MOVL    (SP),ADDRESS                 ; SET NEW "CURRENT" ADDRESS
             00000000'EF   D5  0553   656          TSTL    OUTPUT_FILE                  ; AVOID SUBHEADNG IF NOT A FILE
                     14   13  0559   657          BEQL    59$                          ; BRANCH IF NOT
                          055B   658          SUBHD   <MEMORY DUMP>
                          0568   659          SKIP    PAGE
 08 00000110'EF    05   E0  056F   660 59$:     BBS     #OPT$V_INST,OPTIONS,57$      ; BRANCH IF RANGE OF INSTRUCTIONS
    00000000'EF    03   FB  0577   661          CALLS   #3,DUMP                      ; DUMP MEMORY
                     04   057E   662          RET
                   01A6   31  057F   663 57$:     BRW     150$                         ; GO SHOW RANGE OF DATA ITEMS
                          0582   664
                          0582   665 ;
                          0582   666 ; NO ADDRESS SPECIFIED - SKIP TO NEXT DATA ITEM
                          0582   667 ;
    53   00000140'EF   DE  0582   668 60$:     MOVAL   ADDRESS,R3
    63   00000144'EF   C0  0589   669          ADDL    DATALEN,(R3)                 ; SET "." TO NEXT DATA SEGMENT
                          0590   670 ;
                          0590   671 ; DISPLAY DATA AT "ADDRESS".
                          0590   672 ;
             00000000'EF   D4  0590   673 70$:     CLRL    LINE_COUNT                   ; INHIBIT PAGE EJECTS
    55   00000000'EF   9E  0596   674          MOVAB   BUFFER,R5                    ; ADDRESS OF INSTRUCTION/DATA BUFFER
                          059D   675          ALLOC   110,R2                       ; ALLOCATE SYMBOLIZE OUTPUT BUFFER
 03 00000110'EF    05   E1  05AF   676          BBC     #OPT$V_INST,OPTIONS,71$      ; CONTINUE IF NOT INSTRUCTION
                 00DF   31  05B7   677          BRW     120$                         ; BRANCH IF INSTRUCTION MODE
    00000144'EF    04   D0  05BA   678 71$:     MOVL    #4,DATALEN                   ; SET LENGTH OF DATA SEGMENT
 07 00000110'EF    07   E1  05C1   679          BBC     #OPT$V_TIME,OPTIONS,72$      ; BRANCH IF NOT /TIME
    00000144'EF    08   D0  05C9   680          MOVL    #8,DATALEN                   ; SET LENGTH OF TIME QUADWORD
                          05D0   681 72$:     TRYMEM  @(R3),(R5),DATALEN           ; GET DATA
                   42 50   E9  05E2   682          BLBC    R0,90$                       ; IF NOT AVAILABLE
             000001C0 8F   D3  05E5   683          BITL    #OPT$M_TIME!OPT$M_PSL!OPT$M_COND,-
             00000110'EF          05EB   684                  OPTIONS                      ; TIME, COND OR PSL?
                     4A   12  05F0   685          BNEQ    92$                          ; BRANCH IF EITHER
                     55   DD  05F2   686          PUSHL   R5                           ; ADDRESS OF COPY OF DATA ITEM
                     04   DD  05F4   687          PUSHL   #4                           ; LENGTH OF DATA ITEM
                     51   DD  05F6   688          PUSHL   R1                           ; CONTENTS OF LOCATION
                     52   DD  05F8   689          PUSHL   R2                           ; ADDRESS OF OUTPUT BUFFER
                     63   DD  05FA   690          PUSHL   (R3)                         ; ADDRESS TO TRANSLATE
 00000000'EF         02   FB  05FC   691          CALLS   #2,SYMBOLIZE                 ; TRANSLATE TO SYMBOL+OFFSET
                     61   D5  0603   692          TSTL    (R1)                         ; CHECK LENGTH OF RESULT STRING
                     10   13  0605   693          BEQL    80$                          ; BRANCH IF NOT TRANSLATED
                     51   DD  0607   694          PUSHL   R1                           ; ADDRESS OF STRING DESCRIPTOR
                          0609   695          PRINT   4,<!AS:  !XL   "!AF">
                     04   0616   696          RET
                          0617   697 80$:
                     63   DD  0617   698          PUSHL   (R3)
                          0619   699          PRINT   4,<!XL:  !XL   "!AF">
                     04   0626   700          RET
                          0627   701
                     63   DD  0627   702 90$:     PUSHL   (R3)                         ; ADDRESS
                          0629   703          SIGNAL  1,NOTINPHYS                  ; NOT IN PHYSICAL MEMORY
                     04   063B   704          RET
                          063C   705 ;
                          063C   706 ; EXAMINE /PSL or EXAMINE/TIME or EXAMINE/CONDITION_VALUE
                          063C   707 ;
                          063C   708
```

```
                                 063C   709 92$:
          24 00000110'EF    06   E0 063C   710          BBS     #OPT$V_PSL,OPTIONS,96$  ; BRANCH IF EXAM/PSL
          29 00000110'EF    08   E0 0644   711          BBS     #OPT$V_COND,OPTIONS,98$ ; BRANCH IF EXAM/CONDITION_VALUE
                                 064C   712          $ASCTIM_S TIMLEN = (R2),-
                                 064C   713                   TIMBUF = (R2),-
                                 064C   714                   TIMADR = (R5)           ; FORMAT TIME
                        2B 50    E8 065B   715          BLBS    R0,110$                 ; BRANCH IF SUCCESS
                                 065E   716 95$:
                           50    DD 065E   717          PUSHL   R0                      ; ERROR
              00000000'GF    01    FB 0660   718          CALLS   #1,G^LIB$SIGNAL         ; SIGNAL
                                 04 0667   719          RET
                                 0668   720 96$:
                           52    DD 0668   721          PUSHL   R2                      ; BUFFER ADDRESS
                           65    DD 066A   722          PUSHL   (R5)                    ; PSL ADDRESS
              00000000'EF    02    FB 066C   723          CALLS   #2,FORMAT_PSL           ; FORMAT IT
                           14    11 0673   724          BRB     110$                    ;
                                 0675   725 98$:
                                 0675   726          $GETMSG_S -
                                 0675   727                   MSGID = (R5) -          ; MESSAGE CODE
                                 0675   728                   MSGLEN = (R2) -         ; LENGTH
                                 0675   729                   BUFADR = (R2)           ; ADDRESS
                        D5 50    E9 0686   730          BLBC    R0,95$
                                 0689   731 110$:
                           52    DD 0689   732          PUSHL   R2
                                 068B   733          PRINT   1,<!AS>
                                 04 0698   734          RET
                                 0699   735
                                 0699   736 ;
                                 0699   737 ; EXAMINE/INSTRUCTION
                                 0699   738 ;
                                 0699   739
                                 0699   740 120$:    TRYMEM  @(R3),(R5),#64          ; GET MAXIMUM LENGTH OF INSTRUCTION
                        03 50    E8 06AB   741          BLBS    R0,130$                 ; OK
                         FF76    31 06AE   742          BRW     90$                     ; IF NOT AVAILABLE
                                 06B1   743 130$:    ALLOC   128,R4                  ; BUFFER FOR INSTRUCTION TEXT
                           55    DD 06C3   744          PUSHL   R5                      ; ADDRESS OF INSTRUCTION STREAM
                 000007A1'EF    9F 06C5   745          PUSHAB  INSTR_VALUE             ; ADDRESS OF SYMBOLIZE ROUTINE
                           54    DD 06CB   746          PUSHL   R4                      ; ADDRESS TO RETURN LENGTH
                           54    DD 06CD   747          PUSHL   R4                      ; ADDRESS OF OUTPUT BUFFER DESCRIPTOR
                        0C AE    DF 06CF   748          PUSHAL  12(SP)                  ; ADDRESS OF INSTRUCTION STREAM POINTER
              00000000'GF    04    FB 06D2   749          CALLS   #4,G^LIB$INS_DECODE     ; DECODE INSTRUCTION
                        39 50    E9 06D9   750          BLBC    R0,190$                 ; AND SIGNAL ANY ERRORS
          00000144'EF    8E    55    C3 06DC   751          SUBL3   R5,(SP)+,DATALEN        ; SAVE LENGTH OF INSTRUCTION
                           54    DD 06E4   752          PUSHL   R4                      ; PUSH ADDRESS OF INSTRUCTION TEXT
                           52    DD 06E6   753          PUSHL   R2                      ; ADDRESS OF RESULT BUFFER
                           63    DD 06E8   754          PUSHL   (R3)                    ; ADDRESS TO TRANSLATE
              00000000'EF    02    FB 06EA   755          CALLS   #2,SYMBOLIZE            ; TRANSLATE TO SYMBOL+OFFSET
                           61    D5 06F1   756          TSTL    (R1)                    ; CHECK LENGTH OF RESULT STRING
                           10    13 06F3   757          BEQL    180$                    ; BRANCH IF NOT TRANSLATED
                           51    DD 06F5   758          PUSHL   R1                      ; ADDRESS OF STRING DESCRIPTOR
                                 06F7   759          PRINT   2,<!AS:  !AS>
                                 04 0704   760          RET
                           63    DD 0705   761 180$:    PUSHL   (R3)
                                 0707   762          PRINT   2,<!XL:  !AS>
                                 04 0714   763          RET
                                 0715   764 190$:    SIGNAL  0,NOINSTRAN             ; SIGNAL ANY ERRORS FROM DECODE
                                 04 0727   765          RET
```

```
                                0728   766
                                0728   767 ;
                                0728   768 ; RANGE OF INSTRUCTIONS TO BE DUMPED
                                0728   769 ;
                                0728   770
        00000110'EF    18   CA  0728   771 150$:   BICL    #OPT$M_RANGE.OPT$M_LENGTH,OPTIONS ; INDICATE ONLY ADDRESS
00000040'EF    0000010C'EF   9E  072F   772         MOVAB   EXPR_STACK,ESP            ; POP ALL ARGUMENTS OFF STACK
          55    04 AE    6E   C1  073A   773         ADDL3   (SP),4(SP),R5             ; GET ENDING ADDRESS
              00000144'EF    D4  073F   774         CLRL    DATALEN                   ; DO NOT SKIP FIRST INSTRUCTION
          53   00000140'EF    DE  0745   775         MOVAL   ADDRESS,R3
                                074C   776 ;
                                074C   777 ; ATTEMPT TO LOCATE THE FIRST INSTRUCTION WHICH MAKES SENSE IN THE RANGE.
                                074C   778 ; THIS IS DONE BY ATTEMPTING TO INTERPRET N INSTRUCTIONS IN A ROW WHICH
                                074C   779 ; APPEAR IN THE LIST OF "NORMAL" INSTRUCTIONS.  IF WE CAN'T MAKE SENSE
                                074C   780 ; OUT OF N IN A ROW, THEN WE INCREMENT "." BY 1 BYTE AND TRY AGAIN.
                                074C   781 ; IF THE ENTIRE RANGE IS EXHAUSTED WITHOUT FINDING A VALID SEQUENCE,
                                074C   782 ; THEN GIVE UP AND RESET "." BACK TO THE ORIGINALLY REQUESTED ADDRESS
                                074C   783 ; AND SHOW THE USER THE CRAP.
                                074C   784 ;
             10    04 AE   D1  074C   785         CMPL    4(SP),#16                 ; IS THE PRE-SCAN WORTH IT?
                      36   19  0750   786         BLSS    155$                      ; BRANCH IF NOT
      2E 00000110'EF    09   E0  0752   787         BBS     #OPT$V_NOSKIP,OPTIONS,155$ ; ALLOW USER TO OVERRIDE PRE-SCAN
         000007F3'EF    00   FB  075A   788 152$:   CALLS   #0,TRY_SEQUENCE ; EXAMINE NEXT N INSTRUCTIONS
                 0C    50   E8  0761   789         BLBS    R0,154$                   ; BRANCH IF THEY ARE REASONABLE
                       63   D6  0764   790         INCL    (R3)                      ; SKIP AHEAD ONE BYTE
                 55    63   D1  0766   791         CMPL    (R3),R5                   ; END OF RANGE?
                       EF   1F  0769   792         BLSSU   152$                      ; LOOP IF NOT
                 63    6E   D0  076B   793         MOVL    (SP),(R3)                 ; RESET "." TO ORIGINAL ADDRESS
                       18   11  076E   794         BRB     155$
              7E   63    6E   C3  0770   795 154$:   SUBL3   (SP),(R3),-(SP)         ; NUMBER OF BYTES SKIPPED
                       12   13  0774   796         BEQL    155$                      ; SKIP MESSAGE IF NOTHING SKIPPED
                            0776   797         SIGNAL  1,INSKIPPED               ; INFORM NUMBER OF BYTES SKIPPED
                            0788   798 ;
                            0788   799 ; SHOW THE RANGE OF INSTRUCTIONS, UPDATING "." ON EACH INSTRUCTION
                            0788   800 ;
FD4B CF    00    FB  0788   801 155$:   CALLS   #0,EXAM_MEMORY           ; RECURSIVELY SHOW NEXT DATA ITEM
     55    63    D1  078D   802         CMPL    (R3),R5                   ; END OF RANGE YET?
           F6    1F  0790   803         BLSSU   155$                      ; LOOP UNTIL DONE
     50    01    D0  0792   804         MOVL    #1,R0                     ; SUCCESS
                 04  0795   805         RET
                     0796   806
                     0796   807         .DSABL  LSB
```

```
                                0796   809          .SBTTL  INSTR_VALUE, SYMBOLIZE ADDRESS INTO BUFFER
                                0796   810 ;---
                                0796   811 ;
                                0796   812 ;       OUTPUT A STRING DESCRIBING AN ADDRESS.  THE ADDRESS IS CONVERTED
                                0796   813 ;       TO A "REAL" ADDRESS BY COMPUTING THE OFFSET FROM THE INSTRUCTION
                                0796   814 ;       BUFFER.  THEN THE SYMBOL TABLE IS SEARCHED FOR THE CLOSEST SYMBOL.
                                0796   815 ;       IF NOT FOUND IN THE SYMBOL TABLE, A LONGWORD HEX STRING IS WRITTEN.
                                0796   816 ;
                                0796   817 ; INPUTS:
                                0796   818 ;
                                0796   819 ;       4(AP) = ADDRESS OF LONGWORD CONTAINING ADDRESS
                                0796   820 ;       8(AP) = ADDRESS OF RESULT BUFFER DESCRIPTOR
                                0796   821 ;       12(AP) = ADDRESS OF WORD TO RECEIVE RESULT LENGTH
                                0796   822 ;       16(AP) = ADDRESS OF FLAG: 1 IF ABSOLUTE ADDRESS, 0 IF RELATIVE ADDRESS
                                0796   823 ;
                                0796   824 ; OUTPUTS:
                                0796   825 ;
                                0796   826 ;       THE STRING IS WRITTEN INTO THE RESULT BUFFER.
                                0796   827 ;---
                                0796   828
                                0796   829 XL_STRING:
      4C 58 21 0000079E'010E0000' 0796   830          .ASCID  '!XL'
                                07A1   831
                                07A1   832 INSTR_VALUE:                         ; SYMBOLIZE INSTRUCTION VALUE
                         000C   07A1   833          .WORD   ^M<R2,R3>
                                07A3   834
              52   04 BC    D0  07A3   835          MOVL    @4(AP),R2           ; GET VALUE (ARGUMENT BY REFERENCE)
              53   08 AC    D0  07A7   836          MOVL    8(AP),R3            ; GET ADDRESS OF DESCRIPTOR
                11 10 BC    E8  07AB   837          BLBS    @16(AP),5$          ; BRANCH IF ABSOLUTE ADDRESS
        50    00000000'EF   9E  07AF   838          MOVAB   BUFFER,R0           ; ADDRESS OF INSTRUCTION BUFFER
                    52   50   C2  07B6   839          SUBL    R0,R2               ; GET OFFSET FROM INSTRUCTION
        52    00000140'EF   C0  07B9   840          ADDL    ADDRESS,R2          ; AND COMPUTE "REAL" ADDRESS
                         53   DD  07C0   841 5$:      PUSHL   R3                  ; ADDRESS OF RESULT BUFFER
                         52   DD  07C2   842          PUSHL   R2                  ; VALUE TO BE CONVERTED
  00000000'EF            02   FB  07C4   843          CALLS   #2,SYMBOLIZE        ; CONVERT TO SYMBOL+OFFSET
                         61   D5  07CB   844          TSTL    (R1)                ; ANYTHING RETURNED IN BUFFER?
                         08   13  07CD   845          BEQL    10$                 ; BRANCH IF NOT
           0C BC         61   B0  07CF   846          MOVW    (R1),@12(AP)        ; RETURN LENGTH TO CALLER
              50         01   D0  07D3   847          MOVL    #1,R0
                              04  07D6   848          RET
                 5E      02   C2  07D7   849 10$:     SUBL    #2,SP               ; ADDRESS TO RETURN FAO LENGTH
                         52   DD  07DA   850          PUSHL   R2                  ; ABSOLUTE VALUE
                         53   DD  07DC   851          PUSHL   R3                  ; ADDRESS OF RESULT BUFFER
                   08    AE   DF  07DE   852          PUSHAL  8(SP)               ; ADDRESS OF WORD TO RECEIVE LENGTH
                   B2    AF   9F  07E1   853          PUSHAB  XL_STRING           ; ADDRESS OF FAO CONTROL STRING
  00000000'GF            04   FB  07E4   854          CALLS   #4,G^SYS$FAO        ; CONVERT INTO BUFFER
           0C BC         8E   B0  07EB   855          MOVW    (SP)+,@12(AP)       ; RETURN LENGTH TO CALLER
              50         01   D0  07EF   856          MOVL    #1,R0               ; SUCCESS
                              04  07F2   857          RET
```

```
                        07F3   859          .SBTTL  TRY_SEQUENCE, DETERMINE IF INSTRUCTIONS VALID
                        07F3   860 ;---
                        07F3   861 ;
                        07F3   862 ; SCAN THE CURRENT INSTRUCTION STREAM LOOKING FOR N INSTRUCTIONS IN
                        07F3   863 ; A ROW WHICH "MAKE SENSE", ACCORDING TO A LIST OF "NORMAL" INSTRUCTIONS.
                        07F3   864 ;
                        07F3   865 ; INPUTS:
                        07F3   866 ;
                        07F3   867 ;       ADDRESS = "REAL" ADDRESS OF INSTRUCTION STREAM
                        07F3   868 ;
                        07F3   869 ; OUTPUTS:
                        07F3   870 ;
                        07F3   871 ;       R0 = TRUE IF STREAM REASONABLE, ELSE FALSE
                        07F3   872 ;---
                        07F3   873
                        07F3   874 TRY_SEQUENCE:
                  0038  07F3   875          .WORD   ^M<R3,R4,R5>
                        07F5   876
55  00000000'EF   9E    07F5   877          MOVAB   BUFFER,R5               ; GET ADDRESS OF INSTRUCTION BUFFER
53  00000140'EF   D0    07FC   878          MOVL    ADDRESS,R3              ; GET CURRENT "."
      54    04   D0    0803   879          MOVL    #4,R4                   ; INITIALIZE LOOP COUNT
                        0806   880 10$:     TRYMEM  (R3),(R5),#64           ; GET MAXIMUM LENGTH OF INSTRUCTION
        28 50   E9    0817   881          BLBC    R0,90$                  ; IF NOT AVAILABLE
      50   65   9A    081A   882          MOVZBL  (R5),R0                 ; PICK UP OPCODE
     21 0000'C0   E9    081D   883          BLBC    NORMAL_INSTRS(R0),80$   ; BRANCH IF UNREASONABLE INSTRUCTION
           55   DD    0822   884          PUSHL   R5                      ; ADDRESS OF INSTRUCTION STREAM
           7E   7C    0824   885          CLRQ    -(SP)                   ; NO SYMBOLIZE ROUTINE OR RETLEN
           5E   DD    0826   886          PUSHL   SP                      ; ADDRESS OF NULL BUFFER DESCRIPTOR
        0C AE   DF    0828   887          PUSHAL  12(SP)                  ; ADDRESS OF INSTRUCTION STREAM POINTER
00000000'GF   04   FB    082B   888          CALLS   #4,G^LIB$INS_DECODE     ; DECODE INSTRUCTION
        0D 50   E9    0832   889          BLBC    R0,90$                  ; IF ERROR, RETURN UNREASONABLE
    50  8E   55   C3    0835   890          SUBL3   R5,(SP)+,R0             ; CALCULATE LENGTH OF INSTRUCTION
        53   50   C0    0839   891          ADDL    R0,R3                   ; AND POINT TO NEXT INSTRUCTION
          C7 54   F5    083C   892          SOBGTR  R4,10$                  ; TRY NEXT INSTRUCTION
        50   01   D0    083F   893          MOVL    #1,R0                   ; RETURN SEQUENCE IS REASONABLE
                   04   0842   894 90$:     RET
             50   D4    0843   895 80$:     CLRL    R0                      ; RETURN SEQUENCE UNREASONABLE
                   04   0845   896          RET
```

```
                                0846   898          .SBTTL  SET_PROCESS -- SELECT SPECIFIED PROCESS
                                0846   899 ;---
                                0846   900 ;
                                0846   901 ;       SET_PROCESS
                                0846   902 ;
                                0846   903 ;       SETUP FOR DISPLAY OF A SPECIFIED PROCESS
                                0846   904 ;
                                0846   905 ;   INPUTS:
                                0846   906 ;
                                0846   907 ;       OPTIONS = OPTIONS MASK (ONLY SYSPROC IS CHECKED)
                                0846   908 ;       PROC_NAME = PROCESS NAME
                                0846   909 ;       PROC_INDEX = PROCESS INDEX (MUTALLY EXCLUSIVE)
                                0846   910 ;
                                0846   911 ;   OUTPUTS:
                                0846   912 ;
                                0846   913 ;       PROC_PID = PROCESS PID (0 IF CURRENT PROCESS)
                                0846   914 ;       BADPROC IF NOT FOUND
                                0846   915 ;
                                0846   916 ;       ALL ERRORS ARE SIGNALLED IMMEDIATELY
                                0846   917 ;
                                0846   918 ;---
                                0846   919
                         01FC   0846   920          .ENTRY  SET_PROCESS,-
                                0848   921                  ^M<R2,R3,R4,R5,R6,R7,R8>
                                0848   922
                                0848   923 ;
                                0848   924 ;       IF /SYSTEM SPECIFIED, SET TO "SYSTEM PROCESS"
                                0848   925 ;
      10 00000110'EF    06    E1  0848   926          BBC     #OPT$V_SYSPROC,OPTIONS,10$ ; BRANCH IF NOT SPECIFIED
                           58    01  CE  0850   927          MNEGL   #1,R8                      ; SET TO -1 (SYSTEM)
00000000'EF    00000000'FF    D0  0853   928 5$:      MOVL    @MMG$AL_SYSPCB,PCBADR      ; SET SYSTEM PCB ADDRESS
                           44    11  085E   929          BRB     20$
                                0860   930 ;
                                0860   931 ;       IF PROCESS INDEX GIVEN, OBTAIN PCB ADDRESS FROM VECTOR
                                0860   932 ;
                                0860   933 10$:
                00000118'EF     D5  0860   934          TSTL    PROC_NAME                  ; WAS NAME SPECIFIED
                           5B    12  0866   935          BNEQ    50$                        ; BRANCH IF SO
                                0868   936          REQMEM  @SCH$GL_MAXPIX             ; MAXIMUM INDEX NUMBER
             58 00000130'EF     32  0875   937          CVTWL   PROC_INDEX,R8              ; CHECK IF "SYSTEM PROCESS"
                           D5    19  087C   938          BLSS    5$                         ; BRANCH IF SO
                        51 58    D1  087E   939          CMPL    R8,R1                      ; CHECK IF IN RANGE
                           3D    1A  0881   940          BGTRU   40$                        ; ERROR IF BAD INDEX
                                0883   941          REQMEM  @SCH$GL_PCBVEC             ; VECTOR OF PCB ADDRESSES
                     51 6148    DE  0890   942          MOVAL   (R1)[R8],R1                ; ADDRESS OF POINTER TO PCB
                                0894   943          REQMEM  (R1),PCBADR                ; STORE ADDRESS OF PCB
                                08A4   944 20$:
                                08A4   945          REQMEM  @PCBADR,PCB,#PCB$C_LENGTH ; GET ENTIRE PCB
                        0074    31  08BD   946          BRW     70$
                                08C0   947 40$:
                        00F8    31  08C0   948          BRW     80$                        ; FAILURE
                                08C3   949 ;
                                08C3   950 ;       LOOKUP SPECIFIED NAME IN ALL PCBS
                                08C3   951 ;
                                08C3   952 50$:
                                08C3   953          REQMEM  @SCH$GL_PCBVEC             ; VECTOR OF PCB ADDRESSES
                        56 51    D0  08D0   954          MOVL    R1,R6                      ; SAVE ADDRESS OF VECTOR
```

```
                                                     08D3   955          REQMEM  @SCH$GL_MAXPIX                  ; MAXIMUM PROCESS INDEX
                                  57    51   D0      08E0   956          MOVL    R1,R7                           ; SAVE NUMBER OF SLOTS
                                        58   D4      08E3   957          CLRL    R8                              ; INITIALIZE CURRENT INDEX
                                                     08E5   958 60$:
                                  51    6648 DE      08E5   959          MOVAL   (R6)[R8],R1                     ; ADDRESS OF POINTER TO PCB
                                                     08E9   960          REQMEM  (R1),PCBADR                     ; GET ADDRESS OF PCB
                                                     08F9   961          REQMEM  @PCBADR,PCB,#PCB$C_LENGTH       ; READ ENTIRE PCB
                            50    00000070'EF  9A      0912   962          MOVZBL  PCB+PCB$T_LNAME,R0              ; GET LENGTH OF NAME
                      20    00000071'EF   50   2D      0919   963          CMPC5   R0,PCB+PCB$T_LNAME+1,#^A' ', -
                                  00000118'EF          0921
                                  0000011C'FF          0926   964                  PROC_NAME,@PROC_NAME+4          ; CHECK IF CORRECT PROCESS
                                        07   13      092B   965          BEQL    70$                             ; BRANCH IF YES
                                  B4 58 57   F3      092D   966          AOBLEQ  R7,R8,60$                       ; LOOP UNTIL FOUND
                                       0087  31      0931   967          BRW     80$                             ; FAILURE
                                                     0934   968 70$:
                                  00000118'EF  7C      0934   969          CLRQ    PROC_NAME                       ; CLEAR SO WE DON'T RE-SEARCH
                            51    0000006C'EF  D0      093A   970          MOVL    PCB+PCB$L_PHD,R1                ; ADDRESS OF PROCESS HEADER
                                        3B   13      0941   971          BEQL    75$                             ; BRANCH IF NOT RESIDENT
                                                     0943   972          REQMEM  (R1),PHD,#PHD$C_LENGTH          ; READ ENTIRE PHD
                  00000000'EF   0000006C'EF  D0      0958   973          MOVL    PCB+PCB$L_PHD,PHDADR            ; SAVE PHD ADDRESS
00000000'EF       000000C8'EF     10   28      0963   974          MOVC    #16,PHD+PHD$L_POBR,POBR         ; GET RELOCATION REGISTERS
            00000000'EF    18   00   EF      096F   975          EXTZV   #PHD$V_POLR,#PHD$S_POLR,POLR,POLR
                                  00000000'EF          0977
                                        10   11      097C   976          BRB     79$
                                  00000000'EF  D4      097E   977 75$:     CLRL    PHDADR                          ; SET PHD NOT RESIDENT
                            10    00   6E   00      2C     0984   978          MOVC5   #0,(SP),#0,#16,POBR             ; ZERO RELOCATION REGISTERS
                                  00000000'EF          0989
                           00000130'EF    58   D0      098E   979 79$:     MOVL    R8,PROC_INDEX                   ; SAVE PROCESS INDEX
                  00000138'EF   00000060'EF  D0      0995   980          MOVL    PCB+PCB$L_PID,PROC_PID          ; SAVE PROCESS PID
                                                     09A0   981 ;
                                                     09A0   982 ;        ZERO THE PROC_PID IF THE PROCESS IS THE CURRENT PROCESS.
                                                     09A0   983 ;        THIS AVOIDS UNNECESSARY OVERHEAD ON THE RUNNING SYSTEM DISPLAYS.
                                                     09A0   984 ;
00000138'EF       0000013C'EF     D1   09A0   985          CMPL    CURRENT_PID,PROC_PID            ; IS PROC_PID SAME AS CUR. PROCESS?
                                  06   12      09AB   986          BNEQ    90$                             ; BRANCH IF NOT
                                  00000138'EF  D4      09AD   987          CLRL    PROC_PID                        ; IF SO, ZERO PROC_PID FOR QAST
                                                     09B3   988 90$:     STATUS  SUCCESS
                                        04   09BA   989          RET
                                                     09BB   990 80$:
                                  00000118'EF  7C      09BB   991          CLRQ    PROC_NAME                       ; CLEAR BAD PROCESS NAME
                                                     09C1   992          SIGNAL  0,BADPROC                       ; NO SUCH PROCESS
                                        04   09D3   993          RET
```

```
                        09D4   995          .SBTTL  CURPROC -- SET RELOCATION TO CURRENT PROCESS
                        09D4   996 ;---
                        09D4   997 ;
                        09D4   998 ;       CURPROC
                        09D4   999 ;
                        09D4  1000 ;       THIS ROUTINE SETS UP THE RELOCATION REGISTERS TO
                        09D4  1001 ;       REFER TO THE PROCESS EXECUTING WHEN THE SYSTEM
                        09D4  1002 ;       CRASHED.
                        09D4  1003 ;
                        09D4  1004 ;   INPUTS:
                        09D4  1005 ;
                        09D4  1006 ;       NONE
                        09D4  1007 ;
                        09D4  1008 ;   OUTPUTS:
                        09D4  1009 ;
                        09D4  1010 ;       P0BR-P1LR ARE SETUP FOR ''CRASH'' PROCESS
                        09D4  1011 ;
                        09D4  1012 ;---
                        09D4  1013
                        09D4  1014 CURPROC::
                003C    09D4  1015          .WORD   ^M<R2,R3,R4,R5>
                        09D6  1016
                        09D6  1017          GETMEM  @SCH$GL_CURPCB          ; ADDRESS OF CURRENT PCB
                        09E3  1018          RETIFERR
    00000000'EF   51    D0    09E7  1019          MOVL    R1,PCBADR               ; SAVE ADDRESS OF PCB
53  00000000'EF   9E    09EE  1020          MOVAB   L^PCB,R3                ; ADDRESS OF OWN PCB
                        09F5  1021          GETMEM  (R1),(R3),#PCB$C_LENGTH ; READ ENTIRE PCB
                        0A06  1022          RETIFERR
    00000130'EF   60 A3 32    0A0A  1023          CVTWL   PCB$L_PID(R3),PROC_INDEX ; SAVE PROCESS INDEX
                        0A12  1024 ;        MOVL    PCB$L_PID(R3),PROC_PID  ; SAVE PROCESS PID
    0000013C'EF   60 A3 D0    0A12  1025          MOVL    PCB$L_PID(R3),CURRENT_PID ; SAVE PID OF CURRENT PROCESS
          00000138'EF   D4    0A1A  1026          CLRL    PROC_PID                ; DO NOT SWITCH CONTEXT IN GETMEM
    00000000'EF   6C A3 D0    0A20  1027          MOVL    PCB$L_PHD(R3),PHDADR    ; SAVE PHD ADDRESS
       54 00000000'EF   9E    0A28  1028          MOVAB   L^PHD,R4                ; ADDRESS OF OWN PHD
                        0A2F  1029          GETMEM  @PCB$L_PHD(R3),(R4),#PHD$C_LENGTH  ; READ ENTIRE PHD
                        0A41  1030          RETIFERR
00000000'EF   00C8 C4 10    28    0A45  1031          MOVC    #16,PHD$L_P0BR(R4),P0BR ; GET P0BR - P1LR
     00000000'EF   18 00    EF    0A4F  1032          EXTZV   #PHD$V_P0LR,#PHD$S_P0LR,P0LR,P0LR   ; ELIM. ASTLVL
          00000000'EF       0A57
                        0A5C  1033          GETMEM  @SCH$GL_MAXPIX          ; GET MAXIMUM PROCESS INDEX
                        0A69  1034          RETIFERR
              50   1F   D0    0A6D  1035          MOVL    #31,R0                  ; START AT HIGH BIT
           03 51   50   E0    0A70  1036 10$:     BBS     R0,R1,11$               ; FIND THE HIGHEST SET BIT
                F9 50   F5    0A74  1037          SOBGTR  R0,10$                  ; WE KNOW WE WILL FIND ONE
     00000134'EF   50   01    C1    0A77  1038 11$:     ADDL3   #1,R0,PIX_WIDTH         ; BIT ''N'' SET MEANS THAT PIX WIDTH FOR
                        0A7F  1039                                              ;  EXTENDED PID IS ''N+1'' BITS
                    04  0A7F  1040          RET
```

```
                                0A80  1042          .SBTTL  SHOW_PROCESS -- DISPLAY SPECIFIED PROCESS
                                0A80  1043 ;---
                                0A80  1044 ;
                                0A80  1045 ;       SHOW_PROCESS
                                0A80  1046 ;
                                0A80  1047 ;       DISPLAY SELECTED FIELDS FROM A SPECIFIED PROCESS
                                0A80  1048 ;
                                0A80  1049 ;   INPUTS:
                                0A80  1050 ;
                                0A80  1051 ;       PROC_NAME OR PROC_INDEX CONTAINS WHICH PROCESS
                                0A80  1052 ;
                                0A80  1053 ;   OUTPUTS:
                                0A80  1054 ;
                                0A80  1055 ;       NONE
                                0A80  1056 ;
                                0A80  1057 ;---
                                0A80  1058
                     4C 4C 41   0A80  1059 ALL:    .ASCII  'ALL'
                                0A83  1060
                                0A83  1061 SHOW_PROCESS::
                        003C    0A83  1062          .WORD   ^M<R2,R3,R4,R5>
                                0A85  1063
00000118'EF   20   F7 AF   03   0A85  1064          CMPC5   #3,ALL,#^A' ',PROC_NAME,@PROC_NAME+4
                0000011C'FF
                          08    0A8F  1065          BNEQ    10$                     ; BRANCH IF NOT ALL
              00000000'EF 00    0A94  1066          CALLS   #0,DISPLAY_PROCS        ; DISPLAY ALL PROCESSES
                          04    0A96  1067          RET
                                0A9D  1068 10$:
          FDA3 CF       00    0A9E  1069          CALLS   #0,SET_PROCESS          ; SETUP TO READ PROCESS
                     07 50      0A9E  1070          BLBC    R0,90$                  ; BRANCH IF ERROR
              00000000'EF 00    0AA3  1071          CALLS   #0,DISPLAY_PROCESS      ; DISPLAY INFORMATION
                                0AA6  1072 90$:
                                0AAD  1073          STATUS  SUCCESS
                          04    0AAD  1074          RET
```

```
                                0AB5  1076          .SBTTL  DISPLAY_HELP -- DISPLAY HELP INFORMATION
                                0AB5  1077 ;---
                                0AB5  1078 ;
                                0AB5  1079 ;       DISPLAY_HELP
                                0AB5  1080 ;
                                0AB5  1081 ;       DISPLAY PORTIONS OF THE HELP FILE BASED ON THE TYPE
                                0AB5  1082 ;       OF INFORMATION REQUESTED IN THE HELP COMMAND.
                                0AB5  1083 ;
                                0AB5  1084 ;   INPUTS:
                                0AB5  1085 ;
                                0AB5  1086 ;       AP POINTS TO THE TPARSE BLOCK
                                0AB5  1087 ;
                                0AB5  1088 ;   OUTPUTS:
                                0AB5  1089 ;
                                0AB5  1090 ;       NONE
                                0AB5  1091 ;
                                0AB5  1092 ;---
                                0AB5  1093
                                0AB5  1094          .ENABL  LSB
                                0AB5  1095
                                0AB5  1096 DISPLAY_HELP::
                          0000  0AB5  1097          .WORD   ^M<>
6D  00000AF6'EF   DE            0AB7  1098          MOVAL   HELPHAND,(FP)           ;Establish Condition Handler
      00000000'GF   DF          0ABE  1099          PUSHAL  G^LIB$GET_INPUT         ;Address of input routine
      00000000'EF   DF          0AC4  1100          PUSHAL  HELP_FLAGS              ;Library search & Prompt options
      00000004'EF   7F          0ACA  1101          PUSHAQ  HELP_LIBRARY            ;Address of library name descriptor
              10 AC   DF          0AD0  1102          PUSHAL  TPA$L_TOKENCNT(AP)      ;Address of input line descriptor
                 00   DD          0AD3  1103          PUSHL   #0                      ;Text line width -- Defaulted
      00000000'GF   DF          0AD5  1104          PUSHAL  G^LIB$PUT_OUTPUT        ;Address of output routine
00000000'GF     06   FB          0ADB  1105          CALLS   #6,G^LBR$OUTPUT_HELP    ;Does the obvious...
              09 50   E8          0AE2  1106          BLBS    R0,10$                  ;Any problems?
                 50   DD          0AE5  1107          PUSHL   R0                      ;YES,...
00000000'GF     01   FB          0AE7  1108          CALLS   #1,G^LIB$SIGNAL         ;    Send it to the condition handler
                              0AEE  1109 10$:    STATUS  SUCCESS                 ;All's well that ends well...
                        04    0AF5  1110          RET
                              0AF6  1111
                              0AF6  1112          .DSABL  LSB
                              0AF6  1113 ;
                              0AF6  1114 ;       Condition Handler for Display_Help.
                              0AF6  1115 ;
                              0AF6  1116
                              0AF6  1117          .ENABL  LSB
                              0AF6  1118
                              0AF6  1119 HELPHAND:
                          0000  0AF6  1120          .WORD   ^M<>
   50   04 AC    D0            0AF8  1121          MOVL    4(AP),R0                ;Get address of Signal Array
                              0AFC  1122 ;
                              0AFC  1123 ;       Check to see if this is a SEVERE or FATAL exception...
                              0AFC  1124 ;       if it is, then CHANGE the severity to ERROR and resignal the
                              0AFC  1125 ;       condition,  else, just resignal the condition.
                              0AFC  1126 ;
                 00   ED          0AFC  1127          CMPZV   #STS$V_SEVERITY,-       ;Position of Severity Field
                 03               0AFE  1128                  #STS$S_SEVERITY,-       ;Size of the Severity Field
              04 A0               0AFF  1129                  CHF$L_SIG_NAME(R0),-    ;Base address of comparison field
                 04               0B01  1130                  #STS$K_SEVERE           ;Test value
                 06   12          0B02  1131          BNEQ    10$                     ;Its not a FATAL severity.
                              0B04  1132 ;
```

```
                     0B04  1133 ;       If we are here, then we do have a FATAL severity code in our
                     0B04  1134 ;       condition value, sooo... we must CHANGE the severity to ERROR.
                     0B04  1135 ;
          02   F0   0B04  1136          INSV    #STS$K_ERROR,-          ;The desired severity code.
          00         0B06  1137                  #STS$V_SEVERITY,-       ;Position of Severity Field
          03         0B07  1138                  #STS$S_SEVERITY,-       ;Size of the Severity Field
       04 A0         0B08  1139                  CHF$L_SIG_NAME(R0)      ;Base address of Signal Name
                     0B0A  1140
50  0918 8F   3C     0B0A  1141 10$:     MOVZWL  #SS$_RESIGNAL,R0        ;Send condition to HANDLER.
               04    0B0F  1142          RET
                     0B10  1143
                     0B10  1144          .DSABL  LSB
```

```
                        0B10  1146          .SBTTL  READ_SYMFILE, Read symbols from given file
                        0B10  1147 ;----
                        0B10  1148 ;
                        0B10  1149 ;       READ_SYMFILE
                        0B10  1150 ;
                        0B10  1151 ;       This command can be executed to add symbol definitions
                        0B10  1152 ;       to the working symbol table by reading object modules
                        0B10  1153 ;       containing GSI entries and defining all symbols found.
                        0B10  1154 ;
                        0B10  1155 ;   Inputs:
                        0B10  1156 ;
                        0B10  1157 ;       FILE_DESC - Descriptor of file specification
                        0B10  1158 ;
                        0B10  1159 ;   Outputs:
                        0B10  1160 ;
                        0B10  1161 ;       The symbols are added to the symbol table.
                        0B10  1162 ;
                        0B10  1163 ;----
                        0B10  1164
                        0B10  1165 s,n_default:
              42 54 53 2E  0B10  1166          .ascii  '.STB'
                 00000004  0B14  1167 sym_def_len = .-sym_default
                        0B14  1168
                     003C  0B14  1169          .entry  read_symfile,-
                        0B16  1170                  ^m<r2,r3,r4,r5>
                        0B16  1171
    50   00000128'EF   9E  0B16  1172          movab   file_desc,r0            ; Address of file descriptor
    52   00000000'EF   9E  0B1D  1173          movab   stbf,r2                 ; R2 = Address of FAB
    53   00000000'EF   9E  0B24  1174          movab   stb,r3                  ; R3 = Address of RAB
       34 A2    60   33  0B2B  1175          cvtwb   (r0),fab$b_fns(r2)      ; Set file name length
    2C A2    04 A0   D0  0B2F  1176          movl    4(r0),fab$l_fna(r2)     ; Set file name address
       35 A2    04   90  0B34  1177          movb    #sym_def_len,fab$b_dns(r2)      ; Set default length
    30 A2    D5 AF   9E  0B38  1178          movab   sym_default,fab$l_dna(r2)       ; Set default address
                        0B3D  1179          $open   (r2)                    ; Attempt to open the file
             29 50   E8  0B46  1180          blbs    r0,5$                   ; branch if ok
             2C A2   DD  0B49  1181          pushl   fab$l_fna(r2)           ; Create descriptor of file name
          7E 34 A2   9A  0B4C  1182          movzbl  fab$b_fns(r2),-(sp)
             7E 50   7D  0B50  1183          movq    r0,-(sp)                ; RMS error codes
                50   F0  0B53  1184          insv    #sts$k_warning,-        ; Change severity to warning
       6E 03    00        0B55  1185                  #sts$v_severity,#sts$s_severity,(sp)
             08 AE   9F  0B58  1186          pushab  8(sp)                   ; Address of file name descriptor
50   00031098 8F   D0  0B5B  1187          movl    #shr$_openin!<3@16>,r0  ; "Error opening input file !AS"
                        0B62  1188          signal  1                       ; Signal with 1 FAO argument
                5A   11  0B70  1189          brb     95$                     ; exit with success
                        0B72  1190 5$:      $connect (r3)                   ; Connect for record access
                        0B7B  1191          signal  RMS,(r3)                ; Report any errors
00000000'EF   00   FB  0B8D  1192          calls   #0,rewind_stb           ; Rewind the file
                        0B94  1193 10$:
00000000'EF   00   FB  0B94  1194          calls   #0,get_symbol           ; Get next symbol in file
             13 50   E9  0B9B  1195          blbc    r0,90$                  ; Branch if end of file
             05 A1   DD  0B9E  1196          pushl   obj$l_sym_value(r1)     ; Value of symbol
             0A A1   DF  0BA1  1197          pushal  obj$t_sym_name+1(r1)    ; Address of symbol
          7E 09 A1   9A  0BA4  1198          movzbl  obj$t_sym_name(r1),-(sp); Length of symbol
00000000'EF   03   FB  0BA8  1199          calls   #3,add_symbol           ; Add to symbol table
                E3   11  0BAF  1200          brb     10$
                        0BB1  1201 90$:
                        0BB1  1202          $close  (r2)                    ; Close the file
```

```
                OBBA  1203              signal  RMS,(r2)                 ; Signal any errors
                OBC0  1204 95$:         status  success
 04             OBD3  1205              ret
```

```
                        0BD4  1207          .sbttl  SEARCH_MEMORY
                        0BD4  1208 ;+
                        0BD4  1209 ;       search memory for the longword pattern
                        0BD4  1210 ;
                        0BD4  1211 ; Inputs:
                        0BD4  1212 ;
                        0BD4  1213 ;       expr_stack      -->     pattern
                        0BD4  1214 ;                               end_addr or length
                        0BD4  1215 ;                               start_addr
                        0BD4  1216 ;
                        0BD4  1217 ; Outputs:
                        0BD4  1218 ;
                        0BD4  1219 ;       None
                        0BD4  1220 ;-
                        0BD4  1221          .enabl  lsb
                        0BD4  1222
                01CC    0BD4  1223          .entry  search_memory,^M<r2,r3,r6,r7,r8>
                        0BD6  1224
52  00000110'EF   D0    0BD6  1225          movl    options, r2             ; Get options
53  00000040'EF   D0    0BDD  1226          movl    esp, r3         ; point to expression stack
     56   08 A3   D0    0BE4  1227          movl    8(r3), r6               ; r6 --> start addr
     57   04 A3   D0    0BE8  1228          movl    4(r3), r7               ; r7 --> end addr or length
     03 52    03   E0    0BEC  1229          bbs     #opt$v_range, r2, 10$   ; br if range option
        57    56   C0    0BF0  1230          addl    r6, r7                  ; add start addr to length
        58    63   D0    0BF3  1231 10$:     movl    (r3), r8                ; r8 --> value to search for
                        0BF6  1232 ;
                        0BF6  1233 ;       Search from (r6) to (r7) for r8
                        0BF6  1234 ;
                58    DD  0BF6  1235          pushl   r8
                57    DD  0BF8  1236          pushl   r7
                56    DD  0BFA  1237          pushl   r6
                        0BFC  1238          print   3,<Searching from !XL to !XL for !XL...>
                        0C09  1239 20$:
        57    56    D1  0C09  1240          cmpl    r6, r7                  ; check for end of search
              25    1A  0C0C  1241          bgtru   40$                     ; GTR if range searched
                        0C0E  1242          trymem  (r6)                    ; get contents of location
          14  50    E9  0C17  1243          blbc    r0, 30$                 ; if not an address
51        58    D1  0C1A  1244          cmpl    r8, r1                  ; does this location match?
              0F    12  0C1D  1245          bneq    30$                     ; Br if not
                        0C1F  1246 ;
                        0C1F  1247 ;       Here with a match
                        0C1F  1248 ;
              56    DD  0C1F  1249          pushl   r6                      ; push address of match
                        0C21  1250          print   1,<Match at !XL>        ; print it out
56            04    C0  0C2E  1251 30$:     addl    #4, r6                  ; step on a longword
              D6    11  0C31  1252          brb     20$                     ; go look at the next
                        0C33  1253 40$:
50            01    D0  0C33  1254          movl    #1, r0                  ; O.K. status
                    04  0C36  1255          ret                             ;
                        0C37  1256
                        0C37  1257          .dsabl  lsb
```

```
                     0C37  1259          .sbttl  ECHO
                     0C37  1260 ;+
                     0C37  1261 ;       ECHO last command
                     0C37  1262 ;
                     0C37  1263 ; Inputs:
                     0C37  1264 ;
                     0C37  1265 ;       R2 - RAB address
                     0C37  1266 ;       R3 - TPARSE BLOCK
                     0C37  1267 ;
                     0C37  1268 ;-
                     0C37  1269
                     0C37  1270 ECHO:
           54   DD   0C37  1271          PUSHL   R4                      ; SAVE
     54   3C A2   D0   0C39  1272          MOVL    RAB$L_FAB(R2),R4        ; R2 = ADDRESS OF FAB
00000000'8F   E1   0C3D  1273          BBC     #DEV$V_TRM,-
   3D 40 A4         0C43  1274                  FAB$L_DEV(R4),20$       ; BRANCH IF NOT TERMINAL
00000000'EF   B5   0C46  1275          TSTW    TT_CHAN                 ; CHANNEL ASSIGNED?
           35   13   0C4C  1276          BEQL    20$
                     0C4E  1277
00000000'8F   E1   0C4E  1278          BBC     #TT$V_SCOPE,-
29 00000000'EF         0C54  1279                  DVI_DEVDEPEND,20$       ; IGNORE IF NOT SCOPE
                     0C5A  1280
                     0C5A  1281          $QIOW_S CHAN=TT_CHAN,-
                     0C5A  1282                  FUNC=#IO$_WRITEVBLK,-
                     0C5A  1283                  P1=@TPA$L_STRINGPTR(R3),-
                     0C5A  1284                  P2=TPA$L_STRINGCNT(R3),-
                     0C5A  1285                  P4=#^X00880000          ; BACKSPACE FIRST!
                     0C83  1286 20$:
           54 8ED0   0C83  1287          POPL    R4                      ; RESTORE
              05   0C86  1288          RSB
                     0C87  1289
                     0C87  1290          .END
```

```
$$                              = 00000001
$$.TMP1                         = 00000001
$$.TMP2                         = 00000062
$$T1                            = 00000001
ADDRESS                           00000140 RG    02
ADD_SYMBOL                        ********   X   03
ALL                               00000A80 R     03
ARGS                            = 00000001
BUFFER                            ********   X   03
CHF$L_SIG_NAME                  = 00000004
CURPROC                           000009D4 RG    03
CURRENT_PID                       0000013C R     02
DATALEN                           00000144 R     02
DEFINE_SYMBOL                     000003DF RG    03
DEV$V_TRM                         ********   X   03
DISPLAY_HELP                      00000AB5 RG    03
DISPLAY_PROCESS                   ********   X   03
DISPLAY_PROCS                     ********   X   03
DUMP                              ********   X   03
DVI_DEVCHAR                       ********   X   03
DVI_DEVDEPEND                     ********   X   03
DVI_DEVDEPND2                     ********   X   03
ECHO                              00000C37 R     03
ESP                               00000040 RG    02
EXAM_MEMORY                       000004D8 RG    03
EXIT_COMMAND                      000003CB RG    03
EXPR_STACK                        0000010C R     02
FAB$B_DNS                       = 00000035
FAB$B_FNS                       = 00000034
FAB$L_DFV                       = 00000040
FAB$L_DNA                       = 00000030
FAB$L_FNA                       = 0000002C
FAB$L_STV                       = 0000000C
FILE_DESC                         00000128 RG    02
FORMAT_PSL                        ********   X   03
GETMEM                            ********   X   03
GET_COMMANDS                      00000107 RG    03
GET_INPUT                         ********   X   03
GET_SYMBOL                        ********   X   03
HANDLER                           ********   X   03
HEADING_ROUTINE                   ********   X   03
HELPHAND                          00000AF6 R     03
HELP_FLAGS                        00000000 R     02
HELP_LIBRARY                      00000004 R     02
HLP$M_GROUP                     = 00000004
HLP$M_PROCESS                   = 00000002
HLP$M_SYSTEM                    = 00000008
INDIRECT_COMMAND                  000002D3 RG    03
INDRAB                            ********   X   03
INPUT_BUFFER                      ********   X   03
INPUT_BUF_LEN                     ********   X   03
INPUT_LEN                         ********   X   03
INPUT_RAB                         0000010C RG    02
INSTR_VALUE                       000007A1 R     03
IO$_WRITEVBLK                     ********   X   03
LBR$OUTPUT_HELP                   ********   X   03
LIB$GET_INPUT                     ********   X   03
LIB$INS_DECODE                    ********   X   03
LIB$PUT_OUTPUT                    ********   X   03
LIB$SIGNAL                        ********   X   03
LIB$TPARSE                        ********   X   03
LINE_COUNT                        ********   X   03
LOGRAB                            ********   X   03
LOG_BUFFER                        00000150 R     02
LOG_BUFFER_LENGTH               = 00000055
LOG_FILE                          ********   X   03
MMG$AL_SYSPCB                     ********   X   03
MSG$_BADPROC                      ********   X   03
MSG$_BADSYM                       ********   X   03
MSG$_EOF                          ********   X   03
MSG$_EXITCMD                      ********   X   03
MSG$_INSKIPPED                    ********   X   03
MSG$_NOINSTRAN                    ********   X   03
MSG$_NOTINPHYS                    ********   X   03
MSG$_SUCCESS                      ********   X   03
NEW_PAGE                          ********   X   03
NORMAL_INSTRS                     00000000 R     03
OBJ$L_SYM_VALUE                 = 00000005
OBJ$T_SYM_NAME                  = 00000009
OPT$M_ALL                       = 00000FFF
OPT$M_COND                      = 00000100
OPT$M_INST                      = 00000020
OPT$M_LENGTH                    = 00000010
OPT$M_NOSKIP                    = 00000200
OPT$M_PSL                       = 00000040
OPT$M_RANGE                     = 00000008
OPT$M_TIME                      = 00000080
OPT$V_COND                      = 00000008
OPT$V_INST                      = 00000005
OPT$V_LENGTH                    = 00000004
OPT$V_NOSKIP                    = 00000009
OPT$V_PO                        = 00000000
OPT$V_P1                        = 00000001
OPT$V_PSL                       = 00000006
OPT$V_RANGE                     = 00000003
OPT$V_SYSPROC                   = 00000006
OPT$V_SYSTEM                    = 00000002
OPT$V_TIME                      = 00000007
OPTIONS                           00000110 RG    02
OUTPUT_FILE                       ********   X   03
POBR                              ********   X   03
POLR                              ********   X   03
PCB                               ********   X   03
PCB$C_LENGTH                    = 00000120
PCB$L_PHD                       = 0000006C
PCB$L_PID                       = 00000060
PCB$T_LNAME                     = 00000070
PCBADR                            ********   X   03
PHD                               ********   X   03
PHD$C_LENGTH                    = 0000017C
PHD$L_POBR                      = 000000C8
PHD$S_POLR                      = 00000018
PHD$V_POLR                      = 00000000
PHDADR                            ********   X   03
```

```
PIX_WIDTH                       00000134 RG    02
PRINT                           ******** X     03
PRINT_SYMBOLS                   ******** X     03
PROC_INDEX                      00000130 RG    02
PROC_NAME                       00000118 RG    02
PROC_PID                        00000138 RG    02
PROMPT                          00000100 R     03
PROMPT_LEN                    = 00000005
RAB$L_FAB                     = 0000003C
RAB$L_RBF                     = 00000028
RAB$L_STV                     = 0000000C
RAB$L_UBF                     = 00000024
RAB$W_RSZ                     = 00000022
RAB$W_USZ                     = 00000020
READ_SYMFILE                    00000B14 RG    03
RELOCATE_BASE                   00000114 RG    02
REPEAT_COMMAND                  ******** X     03
REQMEM                          ******** X     03
REWIND_STB                      ******** X     03
RMS$_EOF                        ******** X     03
SAVE_COMMAND                    ******** X     03
SCH$GL_CURPCB                   ******** X     03
SCH$GL_MAXPIX                   ******** X     03
SCH$GL_PCBVEC                   ******** X     03
SDA_KEY                         ******** X     03
SDA_STATE                       ******** X     03
SEARCH_MEMORY                   00000BD4 RG    03
SET_HEADING                     ******** X     03
SET_PROCESS                     00000846 RG    03
SHOW_EXPR                       0000047C RG    03
SHOW_P0                         ******** X     03
SHOW_P1                         ******** X     03
SHOW_PROCESS                    00000A83 RG    03
SHOW_SYMBOL                     000003F3 RG    03
SHOW_SYSTEM                     ******** X     03
SHR$_OPENIN                   = 00001098
SHR$_SYNTAX                   = 000010F8
SMG$_EOF                        ******** X     03
SS$_RESIGNAL                  = 00000918
STB                             ******** X     03
STBF                            ******** X     03
STR$UPCASE                      ******** X     03
STS$K_ERROR                   = 00000002
STS$K_SEVERE                  = 00000004
STS$K_WARNING                 = 00000000
STS$S_SEVERITY                = 00000003
STS$V_SEVERITY                = 00000000
SUB_HEADING                     ******** X     03
SYMBOLIZE                       ******** X     03
SYMBOL_DESC                     00000148 RG    02
SYMBOL_NAME                     00000120 RG    02
SYMBOL_VALUE                    ******** X     03
SYM_DEFAULT                     00000B10 R     03
SYM_DEF_LEN                   = 00000004
SYS$ASCTIM                      ******** GX    03
SYS$CLOSE                       ******** GX    03
SYS$CONNECT                     ******** GX    03
SYS$FAO                         ******** X     03
SYS$GET                         ******** GX    03
SYS$GETMSG                      ******** GX    03
SYS$OPEN                        ******** GX    03
SYS$PUT                         ******** GX    03
SYS$QIOW                        ******** GX    03
TPA$K_COUNTO                  = 00000008
TPA$L_NUMBER                  = 0000001C
TPA$L_STRINGCNT               = 00000008
TPA$L_STRINGPTR               = 0000000C
TPA$L_TOKENCNT                = 00000010
TPA$L_TOKENPTR                = 00000014
TPA$M_ABBREV                  = 00000002
TPARSE_BLOCK                    0000001C R     02
TRYMEM                          ******** X     03
TRY_SEQUENCE                    000007F3 R     03
TT$V_SCOPE                      ******** X     03
TT2$V_EDITING                   ******** X     03
TT_CHAN                         ******** X     03
XL_STRING                       00000796 R     03
```

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	000001A5 (421.)	02 (2.)	NOPIC USR CON REL LCL NOSHR NOEXE RD WRT NOVEC BYTE
COMMANDS	00000C87 (3207.)	03 (3.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC LONG
LITERALS	00000109 (265.)	04 (4.)	NOPIC USR CON REL LCL NOSHR EXE RD NOWRT NOVEC BYTE

Performance indicators

Phase	Page f* lts	CPU Time	Elapsed Time
Initialization	29	00:00:00.05	00:00:01.07
Command processing	153	00:00:00.59	00:00:04.52
Pass 1	528	00:00:13.88	00:00:47.83
Symbol table sort	0	00:00:01.92	00:00:08.76
Pass 2	233	00:00:03.32	00:00:13.17
Symbol table output	23	00:00:00.14	00:00:00.45
Psect synopsis output	2	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	970	00:00:19.94	00:01:15.82

The working set limit was 2250 pages.
124778 bytes (244 pages) of virtual memory were used to buffer the intermediate code.
There were 100 pages of symbol table space allocated to hold 1844 non-local and 112 local symbols.
1290 source lines were read in Pass 1, producing 42 object records in Pass 2.
50 pages of virtual memory were used to define 48 macros.

Macro library statistics

Macro library name	Macros defined
_255DUA28:[SDA.OBJ]SDALIB.MLB;1	13
_255DUA28:[SYS.OBJ]LIB.MLB;1	6
_255DUA28:[SYSLIB]STARLET.MLB;2	25
TOTALS (all libraries)	44

2088 GETS were required to define 44 macros.

There were no errors, warnings or information messages.

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