```
MMM          MMM     SSSSSSSSSSS     GGGGGGGGGGG   FFFFFFFFFFFFFFF      IIIIIIIII     LLL
MMM          MMM     SSSSSSSSSSS     GGGGGGGGGGG   FFFFFFFFFFFFFFF      IIIIIIIII     LLL
MMM          MMM     SSSSSSSSSSS     GGGGGGGGGGG   FFFFFFFFFFFFFFF      IIIIIIIII     LLL
MMMMMM    MMMMMM  SSS             GGG             FFF                     III        LLL
MMMMMM    MMMMMM  SSS             GGG             FFF                     III        LLL
MMMMMM    MMMMMM  SSS             GGG             FFF                     III        LLL
MMM   MMM    MMM  SSS             GGG             FFF                     III        LLL
MMM   MMM    MMM  SSS             GGG             FFF                     III        LLL
MMM   MMM    MMM  SSS             GGG             FFF                     III        LLL
MMM          MMM     SSSSSSSSS    GGG             FFFFFFFFFFF             III        LLL
MMM          MMM     SSSSSSSSS    GGG             FFFFFFFFFFF             III        LLL
MMM          MMM     SSSSSSSSS    GGG             FFFFFFFFFFF             III        LLL
MMM          MMM             SSS  GGG   GGGGGGGGG  FFF                     III        LLL
MMM          MMM             SSS  GGG   GGGGGGGGG  FFF                     III        LLL
MMM          MMM             SSS  GGG   GGGGGGGGG  FFF                     III        LLL
MMM          MMM             SSS  GGG         GGG  FFF                     III        LLL
MMM          MMM             SSS  GGG         GGG  FFF                     III        LLL
MMM          MMM             SSS  GGG         GGG  FFF                     III        LLL
MMM          MMM  SSSSSSSSSSS     GGGGGGGGG     FFF                 IIIIIIIII     LLLLLLLLLLLLLLL
MMM          MMM  SSSSSSSSSSS     GGGGGGGGG     FFF                 IIIIIIIII     LLLLLLLLLLLLLLL
MMM          MMM  SSSSSSSSSSS     GGGGGGGGG     FFF                 IIIIIIIII     LLLLLLLLLLLLLLL
```

```
LL              IIIIII      SSSSSSSS  TTTTTTTTTT    IIIIII    NN      NN    GGGGGGGG
LL              IIIIII      SSSSSSSS  TTTTTTTTTT    IIIIII    NN      NN    GGGGGGGG
LL                II      SS              TT          II      NN      NN  GG
LL                II      SS              TT          II      NN      NN  GG
LL                II      SS              TT          II      NNNN    NN  GG
LL                II      SS              TT          II      NNNN    NN  GG
LL                II        SSSSSS        TT          II      NN  NN  NN  GG
LL                II        SSSSSS        TT          II      NN  NN  NN  GG
LL                II              SS      TT          II      NN    NNNN  GG  GGGGGG
LL                II              SS      TT          II      NN    NNNN  GG  GGGGGG
LL                II              SS      TT          II      NN      NN  GG      GG      ....
LL                II              SS      TT          II      NN      NN  GG      GG      ....
LLLLLLLLLL      IIIIII    SSSSSSSS        TT        IIIIII    NN      NN    GGGGGG        ....
LLLLLLLLLL      IIIIII    SSSSSSSS        TT        IIIIII    NN      NN    GGGGGG        ....


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
```

```
;     1          0001  0 MODULE listing (IDENT = 'V04-000') =
;     2          0002  1 BEGIN
;     3          0003  1
;     4          0004  1 !
;     5          0005  1 !**************************************************************************
;     6          0006  1 !*                                                                        *
;     7          0007  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                              *
;     8          0008  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                *
;     9          0009  1 !*  ALL RIGHTS RESERVED.                                                  *
;    10          0010  1 !*                                                                        *
;    11          0011  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
;    12          0012  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    13          0013  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  O HER  *
;    14          0014  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY  *
;    15          0015  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY  *
;    16          0016  1 !*  TRANSFERRED.                                                           *
;    17          0017  1 !*                                                                        *
;    18          0018  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE  *
;    19          0019  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT  *
;    20          0020  1 !*  CORPORATION.                                                           *
;    21          0021  1 !*                                                                        *
;    22          0022  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
;    23          0023  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;    24          0024  1 !*                                                                        *
;    25          0025  1 !*                                                                        *
;    26          0026  1 !**************************************************************************
;    27          0027  1
;    28          0028  1 !++
;    29          0029  1 ! FACILITY:  Message compiler
;    30          0030  1 !
;    31          0031  1 ! ABSTRACT:
;    32          0032  1 !
;    33          0033  1 !       This compiler translated message definition language
;    34          0034  1 !       into object modules
;    35          0035  1 !
;    36          0036  1 ! ENVIRONMENT:
;    37          0037  1 !
;    38          0038  1 !       VAX/VMS operating system. unprivileged user mode,
;    39          0039  1 !
;    40          0040  1 ! AUTHOR:  Tim Halvorsen, Nov 1979
;    41          0041  1 !
;    42          0042  1 ! Modified by:
;    43          0043  1 !
;    44          0044  1 !       v03-003 GJA0095         Greg Awdziewicz         10-Aug-1984
;    45          0045  1 !               - Change version # of listing title to v04-00.
;    46          0046  1 !
;    47          0047  1 !       v03-002 GJA0052         Greg Awdziewicz         14-Oct-1983
;    48          0048  1 !               Echo original command line.
;    49          0049  1 !
;    50          0050  1 !       001     JWT0065         Jim Teague              15-Nov-1982
;    51          0051  1 !               Make lib$lp_lines a G^ call.
;    52          0052  1 !
;    53          0053  1 !--
;    54          0054  1
;    55          0055  1 !
;    56          0056  1 ! Include files
;    57          0057  1 !
```

```
;    58          0058  1
;    59          0059  1 LIBRARY 'SYS$LIBRARY:STARLET';          ! VAX/VMS common definitions
;    60          0060  1
;    61          0061  1 LIBRARY 'SYS$LIBRARY:TPAMAC';          ! TPARSE definitions
;    62          0062  1
;    63          0063  1 REQUIRE 'SRC$:MSG.REQ';                ! Command definitions
```

```
;    65          0301  1 %SBTTL 'Routine and Data declarations'
;    66          0302  1 !
;    67          0303  1 ! Table of contents
;    68          0304  1 !
;    69          0305  1
;    70          0306  1 FORWARD ROUTINE
;    71          0307  1     line_with_value,                ! Output a line with a hex value
;    72          0308  1     echo_record,                    ! Echo the input record on one line
;    73          0309  1     syntax_error,                   ! Signal a syntax error
;    74          0310  1     set_line_number,                ! Set line number into buffer
;    75          0311  1     new_page,                       ! Start a new page
;    76          0312  1     put_line,                       ! Output a listing record
;    77          0313  1     error_summary,                  ! Give error summary
;    78          0314  1     end_listing;                    ! Cleanup listing
;    79          0315  1
;    80          0316  1 !
;    81          0317  1 ! Macros
;    82          0318  1 !
;    83          0319  1
;    84          0320  1 MACRO
;    85        M 0321  1     descriptor (string) =           ! Create descriptor of static string
;    86          0322  1         UPLIT(%CHARCOUNT(string),UPLIT BYTE (string))%;
;    87          0323  1
;    88          0324  1 !
;    89          0325  1 ! Literals
;    90          0326  1 !
;    91          0327  1
;    92          0328  1 LITERAL
;    93          0329  1     code_offset = 25,               ! Column where message code starts
;    94          0330  1     linenum_offset = 35,            ! Column where line number starts
;    95          0331  1     record_offset = 40,             ! Column where record starts
;    96          0332  1     line_size = 133;                ! Length of output record buffer
;    97          0333  1
;    98          0334  1 !
;    99          0335  1 ! Storage definitions
;   100          0336  1 !
;   101          0337  1
;   102          0338  1 OWN
;   103          0339  1     page_number:        INITIAL(0),     ! Current page number
;   104          0340  1     line_number:        INITIAL(9999),  ! Current line number (force new page)
;   105          0341  1     lines_per_page:     INITIAL(60),    ! Maximum lines per page
;   106          0342  1     last_echoed:        INITIAL(0),     ! Last input record echoed
;   107          0343  1     errors.             INITIAL(0),     ! # errors detected
;   108          0344  1     warnings:           INITIAL(0),     ! # warnings detected
;   109          0345  1     infos:              INITIAL(0);     ! # informational msgs issued
;   110          0346  1
;   111          0347  1 !
;   112          0348  1 ! External storage
;   113          0349  1 !
;   114          0350  1
;   115          0351  1 EXTERNAL
;   116          0352  1     command_line_desc:  BBLOCK,         ! Descriptor of original command line.
;   117          0353  1     cli_flags:          BITVECTOR,      ! CLI qualifier bits
;   118          0354  1     worst_error:        BBLOCK,         ! Worst error encountered
;   119          0355  1     title_text:         VECTOR,         ! Title text string
;   120          0356  1     module_name:        VECTOR,         ! Descriptor of module name string
;   121          0357  1     input_record:       VECTOR,         ! Descriptor of record into tparse
```

```
;  122     0358  1      input_linenum,                  ! Input line number
;  123     0359  1      listing_fab:        BBLOCK,     ! Listing file FAB
;  124     0360  1      listing_rab:        BBLOCK,     ! Listing file RAB
;  125     0361  1      output_fab:         BBLOCK,     ! Output file FAB
;  126     0362  1      output_rab:         BBLOCK,     ! Output file RAB
;  127     0363  1      input_fab:          BBLOCK,     ! Input file FAB
;  128     0364  1      input_rab:          BBLOCK;     ! Input file RAB
;  129     0365  1
;  130     0366  1 !
;  131     0367  1 ! External routines
;  132     0368  1 !
;  133     0369  1
;  134     0370  1 EXTERNAL ROUTINE
;  135     0371  1      lib$lp_lines: addressing_mode(general),     ! Determine physical lines/page
;  136     0372  1      rms_error;                                  ! Signal RMS-type error
;  137     0373  1
;  138     0374  1 ROUTINE null: NOVALUE = ;


                                                .TITLE  LISTING
                                                .IDENT  \V04-000\

                                                .PSECT  $OWN$,NOEXE,2

                    00000000  00000 PAGE_NUMBER:
                                                .LONG   0                       ;
                    0000270F  00004 LINE_NUMBER:
                                                .LONG   9999                    ;
                    0000003C  00008 LINES_PER_PAGE:
                                                .LONG   60                      ;
                    00000000  0000C LAST_ECHOED:
                                                .LONG   0                       ;
                    00000000  00010 ERRORS:     .LONG   0                       ;
                    00000000  00014 WARNINGS:
                                                .LONG   0                       ;
                    00000000  00018 INFOS:      .LONG   0                       ;

                                                .EXTRN  COMMAND_LINE_DESC
                                                .EXTRN  CLI_FLAGS, WORST_ERROR
                                                .EXTRN  TITLE_TEXT, MODULE_NAME
                                                .EXTRN  INPUT_RECORD, INPUT_LINENUM
                                                .EXTRN  LISTING_FAB, LISTING_RAB
                                                .EXTRN  OUTPUT_FAB, OUTPUT_RAB
                                                .EXTRN  INPUT_FAB, INPUT_RAB
                                                .EXTRN  LIB$LP_LINES, RMS_ERROR

                                                .PSECT  $CODE$,NOWRT,2

                        0000 00000 NULL:        .WORD   Save nothing            ; 0374
                          04 00002              RET                             ;

; Routine Size:  3 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  140          0375  1 %SBTTL 'LINE_WITH_VALUE - Output a listing line'
;  141          0376  1 GLOBAL ROUTINE line_with_value (value) =
;  142          0377  1
;  143          0378  1 !---
;  144          0379  1 !
;  145          0380  1 !       This routine outputs a listing line for a successfully
;  146          0381  1 !       parsed input record.
;  147          0382  1 !
;  148          0383  1 ! Inputs:
;  149          0384  1 !
;  150          0385  1 !       value = Value to be output as hex value on line
;  151          0386  1 !
;  152          0387  1 ! Outputs:
;  153          0388  1 !
;  154          0389  1 !       None
;  155          0390  1 !---
;  156          0391  1
;  157          0392  2 BEGIN
;  158          0393  2
;  159          0394  2 LOCAL
;  160          0395  2     input_length,                       ! Length of input line
;  161          0396  2     buffer:     VECTOR [line_size,BYTE],        ! Output buffer
;  162          0397  2     tmpdesc:    VECTOR [2],             ! Temporary FAO descriptor
;  163          0398  2     bufdesc:    VECTOR [2];             ! Descriptor of buffer
;  164          0399  2
;  165          0400  2 IF NOT .cli_flags [qual_listing]        ! If not /LISTING
;  166          0401  2 THEN
;  167          0402  2     RETURN true;                        ! then do nothing
;  168          0403  2
;  169          0404  2 CH$FILL(' ',record_offset,buffer);      ! Clear beginning of line
;  170          0405  2
;  171          0406  2 set_line_number (buffer);               ! Set line number into buffer
;  172          0407  2
;  173          0408  2 input_length = MINU(.input_rab [rab$w_rsz], line_size-record_offset);
;  174          0409  2
;  175          0410  2 bufdesc [0] = .input_length + record_offset; ! Actual length of line
;  176          0411  2 bufdesc [1] = buffer;                    ! Set buffer address
;  177          0412  2
;  178          0413  2 CH$MOVE(.input_length, .input_rab [rab$l_rbf], buffer+record_offset);
;  179          0414  2
;  180          0415  2 tmpdesc [0] = 8;                        ! Length of FAO output
;  181          0416  2 tmpdesc [1] = buffer + code_offset;     ! Address to put FAO output
;  182          0417  2 $FAO(descriptor('!XL'),0, tmpdesc, .value);
;  183          0418  2
;  184          0419  2 !
;  185          0420  2 !       Output the completed line
;  186          0421  2 !
;  187          0422  2
;  188          0423  2 put_line(bufdesc);                      ! Output the line
;  189          0424  2
;  190          0425  2 RETURN true;
;  191          0426  2
;  192          0427  1 END;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2
```

```
                                 4C  58  21  00000 P.AAB:  .ASCII  \!XL\                                    ;
                                             00003         .BLKB   1
                                   00000003  00004 P.AAA:  .LONG   3                                       ;
                                   00000000' 00008         .ADDRESS P.AAB                                 ;

                                                           .EXTRN  SYS$FAO

                                                           .PSECT  $CODE$,NOWRT,2

                                       003C 00000          .ENTRY  LINE_WITH_VALUE, Save R2,R3,R4,R5      ; 0376
                              5E    FF68   CE   9E 00002   MOVAB   -152(SP), SP                           ;
                              52  0000G    CF   E9 00007   BLBC    CLI_FLAGS, 2$                          ; 0400
        28          20        6E           00   2C 0000C   MOVC5   #0, (SP), #32, #40, BUFFER             ; 0404
                                       10   AE        00011                                              ;
                                       10   AE   9F 00013   PUSHAB  BUFFER                               ; 0406
                        0000V  CF              01   FB 00016   CALLS   #1, SET_LINE_NUMBER               ;
                              50  0000G    CF   3C 0001B   MOVZWL  INPUT_RAB+34, R0                       ; 0408
                        005D  8F               50   B1 00020   CMPW    R0, #93                          ;
                                               04   1B 00025   BLEQU   1$                               ;
                              50      5D       8F   9A 00027   MOVZBL  #93, R0                           ;
                              6E      28       A0   9E 0002B 1$: MOVAB   40(R0), BUFDESC                ; 0410
                           04 AE      10       AE   9E 0002F   MOVAB   BUFFER, BUFDESC+4                 ; 0411
                     38 AE  0000G     DF            50   28 00034   MOVC3   INPUT_LENGTH, @INPUT_RAB+40, BUFFER+40 ; 0413
                        08 AE                  08   D0 0003B   MOVL    #8, TMPDESC                      ; 0415
                        0C AE            29    AE   9E 0003F   MOVAB   BUFFER+25, TMPDESC+4             ; 0416
                                         04    AC   DD 00044   PUSHL   VALUE                            ; 0417
                                         0C    AE   9F 00047   PUSHAB  TMPDESC                          ;
                                               7E   D4 0004A   CLRL    -(SP)                            ;
                                      0000'    CF   9F 0004C   PUSHAB  P.AAA                            ;
                  00000000G  00                  04   FB 00050   CALLS   #4, SYS$FAO                   ;
                                               5E   DD 00057   PUSHL   SP                               ; 0423
                        0000V  CF              01   FB 00059   CALLS   #1, PUT_LINE                     ;
                              50               01   D0 0005E 2$: MOVL    #1, R0                         ; 0425
                                                    04 00061   RET                                      ; 0427

; Routine Size:  98 bytes,    Routine Base:  $CODE$ + 0003
```

```
;  194     0428  1 %SBTTL 'SYNTAX_ERROR'
;  195     0429  1 GLOBAL ROUTINE syntax_error (tparse, message, faoargs) =
;  196     0430  1
;  197     0431  1 !---
;  198     0432  1 !
;  199     0433  1 !       This routine reports a syntax error to the listing
;  200     0434  1 !       file when an error is detected in the input stream.
;  201     0435  1 !
;  202     0436  1 ! Inputs:
;  203     0437  1 !
;  204     0438  1 !       message = Message code describing error
;  205     0439  1 !       tparse = Address of TPARSE block
;  206     0440  1 !
;  207     0441  1 ! Outputs:
;  208     0442  1 !
;  209     0443  1 !       r0 = message code given (so that it can be returned)
;  210     0444  1 !---
;  211     0445  1
;  212     0446  2 BEGIN
;  213     0447  2
;  214     0448  2 BUILTIN
;  215     0449  2     ACTUALCOUNT;                        ! Number of actual arguments
;  216     0450  2
;  217     0451  2 MAP
;  218     0452  2     message:    BBLOCK,                 ! Get at fields of message
;  219     0453  2     tparse:     REF BBLOCK;             ! Address of TPARSE parameter block
;  220     0454  2
;  221     0455  2 LOCAL
;  222     0456  2     bufdesc:    VECTOR [2],             ! Buffer descriptor
;  223     0457  2     buffer: VECTOR [line_size,BYTE],    ! Line buffer
;  224     0458  2     pos,                                ! Position to put indicator on line
;  225     0459  2     status;                             ! Status code
;  226     0460  2
;  227     0461  2 !
;  228     0462  2 !       Increment the number of messages for each type of severity
;  229     0463  2 !
;  230     0464  2
;  231     0465  2 IF .message [sts$v_severity] EQL sts$k_warning
;  232     0466  2 THEN
;  233     0467  2     warnings = .warnings + 1
;  234     0468  2 ELSE
;  235     0469  2     IF .message [sts$v_severity] EQL sts$k_info
;  236     0470  2 THEN
;  237     0471  2     infos = .infos + 1
;  238     0472  2 ELSE
;  239     0473  2     errors = .errors + 1;
;  240     0474  2
;  241     0475  2 !
;  242     0476  2 !       Save worst error encountered during compile
;  243     0477  2 !
;  244     0478  2
;  245     0479  2 IF .message [sts$v_severity] GEQ .worst_error [sts$v_severity]
;  246     0480  2 THEN
;  247     0481  2     worst_error = .message OR sts$m_inhib_msg;
;  248     0482  2
;  249     0483  2 !
;  250     0484  2 !       Echo the input record first
```

```
;  251     0485  2 !
;  252     0486  2
;  253     0487  2 IF .cli_flags [qual_listing]            ! If /LISTING specified,
;  254     0488  2 THEN
;  255     0489  3     BEGIN
;  256     0490  3     LOCAL save;
;  257     0491  3     save =  last_echoed;                ! Save line number last echoed
;  258     0492  3     echo_record();                      ! Echo the input record on one line
;  259     0493  3     last_echoed = .save;                ! Restore so call below will not no-op
;  260     0494  2     END;
;  261     0495  2
;  262     0496  2 echo_record(output_rab);                ! also to sys$output as well
;  263     0497  2
;  264     0498  2 !
;  265     0499  2 !       Figure out the position to put the indicator marker and output indicator
;  266     0500  2 !
;  267     0501  2
;  268     0502  2 pos = 1;                                ! Set column to 1
;  269     0503  2 INCR i FROM 0 TO .tparse [tpa$l_tokenptr] - 1 - .input_record [1]
;  270     0504  2 DO
;  271     0505  2     IF .(.input_record [1] + .i)<0,8> EQL '     ' ! If TAB detected,
;  272     0506  2     THEN
;  273     0507  3         pos = .pos + 9 - (.pos MOD 8)   ! Increment column to tab stop
;  274     0508  2     ELSE
;  275     0509  2         pos = .pos + 1;                 ! Else, increment column by one
;  276     0510  2
;  277     0511  2 pos = record_offset + .pos-1;           ! Figure column in output listing
;  278     0512  2
;  279     0513  2 IF (.pos GTRU line_size) OR             ! If off end of line
;  280     0514  3    (.tparse [tpa$l_stringcnt] EQLU 0)   ! or end or record,
;  281     0515  2 THEN
;  282     0516  2     pos = record_offset;                ! then just put at start of line
;  283     0517  2
;  284     0518  2 bufdesc [0] = .pos + 1;                 ! Size of buffer
;  285     0519  2 bufdesc [1] = buffer;                   ! Address of buffer
;  286     0520  2
;  287     0521  2 CH$FILL (' ', .pos, buffer);            ! Clear buffer
;  288     0522  2
;  289     0523  2 buffer [.pos] = '!';                    ! Insert indicator
;  290     0524  2
;  291     0525  2 put_line (bufdesc);                     ! Output indicator
;  292     0526  2
;  293     0527  2 !
;  294     0528  2 !       Get text of error message
;  295     0529  2 !
;  296     0530  2
;  297     0531  2 bufdesc [0] = line_size;                ! Size of buffer
;  298     0532  2
;  299     0533  2 $GETMSG (MSGID=.message,BUFADR=bufdesc,MSGLEN=bufdesc);
;  300     0534  2
;  301     0535  2 IF ACTUALCOUNT() GTR 2                  ! If FAO arguments specified,
;  302     0536  2 THEN
;  303     0537  3     BEGIN
;  304     0538  3     LOCAL
;  305     0539  3         ctrbuffer: VECTOR [line_size,BYTE],     ! Control string
;  306     0540  3         ctrdesc: VECTOR [2];            ! Control string descriptor
;  307     0541  3
```

```
;  308          0542  3     ctrdesc [0] = .bufdesc [0];          ! Copy FAO string descriptor
;  309          0543  3     ctrdesc [1] = ctrbuffer;
;  310          0544  3     CH$MOVE(.bufdesc [0], .bufdesc [1], ctrbuffer); ! Save string
;  311          0545  3     bufdesc [0] = line_size;             ! Reset output buffer size
;  312          0546  3     $FAOL(CTRSTR=ctrdesc,OUTLEN=bufdesc,OUTBUF=bufdesc,PRMLST=faoargs);
;  313          0547  2     END;
;  314          0548  2
;  315          0549  2 !
;  316          0550  2 !       Output the message
;  317          0551  2 !
;  318          0552  2
;  319          0553  2 put_line (bufdesc);                      ! Output message buffer
;  320          0554  2 put_line (bufdesc, output_rab);          ! and also to sys$output as well
;  321          0555  2
;  322          0556  2 RETURN .message OR sts$m_inhib_msg;      ! return with input status
;  323          0557  2                                          ! and mark already signaled
;  324          0558  2
;  325          0559  1 END;


                                                        .EXTRN  SYS$GETMSG, SYS$FAOL

                                        01FC 00000      .ENTRY  SYNTAX_ERROR, Save R2,R3,R4,R5,R6,R7,R8      ; 0429
                                58      0000V  CF  9E 00002     MOVAB   PUT_LINE, R8
                                57      0000'  CF  9E 00007     MOVAB   LAST_ECHOED, R7
                                5E      FEEC   CE  9E 0000C     MOVAB   -280(SP), SP
        50      08  AC          03             00  EF 00011     EXTZV   #0, #3, MESSAGE, R0                  ; 0465
                                               05  12 00017     BNEQ    1$
                                  08           A7  D6 00019     INCL    WARNINGS                             ; 0467
                                               0D  11 0001C     BRB     3$
                                03             50  D1 0001E 1$: CMPL    R0, #3                               ; 0469
                                               05  12 00021     BNEQ    2$
                                  0C           A7  D6 00023     INCL    INFOS                                ; 0471
                                               03  11 00026     BRB     3$
                                  04           A7  D6 00028 2$: INCL    ERRORS                               ; 0473
        50      0000G   CF          03             00  ED 0002B 3$: CMPZV   #0, #3, WORST_ERROR, R0          ; 0479
                                               0B  14 00032     BGTR    4$
                0000G   CF      08      AC 10000000    8F  C9 00034     BISL3   #268435456, MESSAGE, WORST_ERROR  ; 0481
                                0B      0000G  CF  E9 0003F 4$: BLBC    CLI_FLAGS, 5$                        ; 0487
                                52             67  D0 00044     MOVL    LAST_ECHOED, SAVE                    ; 0491
                            0000V  CF          00  FB 00047     CALLS   #0, ECHO_RECORD                      ; 0492
                                67             52  D0 0004C     MOVL    SAVE, LAST_ECHOED                    ; 0493
                                         0000G  CF  9F 0004F 5$: PUSHAB  OUTPUT_RAB                           ; 0496
                            0000V  CF          01  FB 00053     CALLS   #1, ECHO_RECORD
                                56             01  D0 00058     MOVL    #1, POS                              ; 0502
                                52       04    AC  D0 0005B     MOVL    TPARSE, R2                           ; 0503
                    53      14      A2 0000G   CF  C3 0005F     SUBL3   INPUT_RECORD+4, 20(R2), R3
                                50             01  CE 00066     MNEGL   #1, I
                                               1E  11 00069     BRB     8$
                                09      0000GDF 40  91 0006B 6$: CMPB    @INPUT_RECORD+4[I], #9               ; 0505
                                               14  12 00071     BNEQ    7$
        7E          00          56             01  7A 00073     EMUL    #1, POS, #0, -(SP)                   ; 0507
        51          51          8E             08  7B 00078     EDIV    #8, (SP)+, R1, R1
                    51          56             51  C3 0007D     SUBL3   R1, POS, R1
                                56      09     A1  9E 00081     MOVAB   9(R1), POS
                                               02  11 00085     BRB     8$
```

```
                                       56 D6 00087 7$:    INCL    POS                                  ; 0509
                DE           50        53 F2 00089 8$:    AOBLSS  R3, I, 6$                            ; 0505
                             56        27 C0 0008D        ADDL2   #39, POS                             ; 0511
                  00000085   8F        56 D1 00090        CMPL    POS, #133                            ; 0513
                                       05 1A 00097        BGTRU   9$                                   ;
                                  08   A2 D5 00099        TSTL    8(R2)                                ; 0514
                                       03 12 0009C        BNEQ    10$                                  ;
                             56        28 D0 0009E 9$:    MOVL    #40, POS                             ; 0516
                        F8   AD   01   A6 9E 000A1 10$:   MOVAB   1(R6), BUFDESC                       ; 0518
                        FC   AD FF70   CD 9E 000A6        MOVAB   BUFFER, BUFDESC+4                    ; 0519
        56      20           6E        00 2C 000AC        MOVC5   #0, (SP), #32, POS, BUFFER           ; 0521
                                FF70   CD    000B1                                                     ;
                     FF70 CD46         21 90 000B4        MOVB    #33, BUFFER[POS]                     ; 0523
                                  F8   AD 9F 000BA        PUSHAB  BUFDESC                              ; 0525
                             68        01 FB 000BD        CALLS   #1, PUT_LINE                         ;
                        F8   AD   85   8F 9A 000C0        MOVZBL  #133, BUFDESC                        ; 0531
                             7E        0F 7D 000C5        MOVQ    #15, -(SP)                           ; 0533
                                  F8   AD 9F 000C8        PUSHAB  BUFDESC                              ;
                                  F8   AD 9F 000CB        PUSHAB  BUFDESC                              ;
                                  08   AC DD 000CE        PUSHL   MESSAGE                              ;
                  00000000G  00        05 FB 000D1        CALLS   #5, SYS$GETMSG                       ;
                             02        6C 91 000D8        CMPB    (AP), #2                             ; 0535
                                       28 1B 000DB        BLEQU   11$                                  ;
                             6E   F8   AD D0 000DD        MOVL    BUFDESC, CTRDESC                     ; 0542
                        04   AE   08   AE 9E 000E1        MOVAB   CTRBUFFER, CTRDESC+4                 ; 0543
             08   AE    FC   BD   F8   AD 28 000E6        MOVC3   BUFDESC, @BUFDESC+4, CTRBUFFER       ; 0544
                        F8   AD   85   8F 9A 000ED        MOVZBL  #133, BUFDESC                        ; 0545
                                  0C   AC 9F 000F2        PUSHAB  FAOARGS                              ; 0546
                                  F8   AD 9F 000F5        PUSHAB  BUFDESC                              ;
                                  F8   AD 9F 000F8        PUSHAB  BUFDESC                              ;
                                  0C   AE 9F 000FB        PUSHAB  CTRDESC                              ;
                  00000000G  00        04 FB 000FE        CALLS   #4, SYS$FAOL                         ;
                                  F8   AD 9F 00105 11$:   PUSHAB  BUFDESC                              ; 0553
                             68        01 FB 00108        CALLS   #1, PUT_LINE                         ;
                               0000G   CF 9F 0010B        PUSHAB  OUTPUT_RAB                           ; 0554
                                  F8   AD 9F 0010F        PUSHAB  BUFDESC                              ;
                             68        02 FB 00112        CALLS   #2, PUT_LINE                         ;
                50      08   AC 10000000  8F C9 00115     BISL3   #268435456, MESSAGE, R0              ; 0556
                                          04 0011E        RET                                          ; 0559

; Routine Size:  287 bytes,    Routine Base:  $CODE$ + 0065
```

```
;  327          0560  1 %SBTTL 'ECHO_RECORD'
;  328          0561  1 GLOBAL ROUTINE echo_record (rab) =
;  329          0562  1
;  330          0563  1 !---
;  331          0564  1 !
;  332          0565  1 !       This routine simply outputs the input record at the
;  333          0566  1 !       correct position on the output listing.
;  334          0567  1 !
;  335          0568  1 ! Inputs:
;  336          0569  1 !
;  337          0570  1 !       rab = Address of RAB to use (if not specified, use listing_rab)
;  338          0571  1 !       input_rab = Address of input file RAB
;  339          0572  1 !
;  340          0573  1 ! Outputs:
;  341          0574  1 !
;  342          0575  1 !       None
;  343          0576  1 !---
;  344          0577  1
;  345          0578  2 BEGIN
;  346          0579  2
;  347          0580  2 BUILTIN
;  348          0581  2     NULLPARAMETER;                      ! True if not specified
;  349          0582  2
;  350          0583  2 LOCAL
;  351          0584  2     input_length,                       ! Length of input line
;  352          0585  2     buffer:     VECTOR [line_size,BYTE],        ! Line buffer
;  353          0586  2     bufdesc:    VECTOR [2];                     ! Buffer descriptor
;  354          0587  2
;  355          0588  2 IF .input_linenum EQL .last_echoed       ! If already echoed,
;  356          0589  2 THEN
;  357          0590  2     RETURN true;                        ! then skip it
;  358          0591  2 last_echoed = .input_linenum;           ! Remember last line echoed
;  359          0592  2
;  360          0593  2 CH$FILL(' ',record_offset,buffer);      ! Clear beginning of line
;  361          0594  2 set_line_number (buffer);               ! Set line number into buffer
;  362          0595  2
;  363          0596  2 input_length = MINU(.input_rab [rab$w_rsz], line_size-record_offset);
;  364          0597  2
;  365          0598  2 bufdesc [0] = .input_length + record_offset; ! Actual length of line
;  366          0599  2 bufdesc [1] = buffer;                   ! Set buffer address
;  367          0600  2
;  368          0601  2 CH$MOVE(.input_length, .input_rab [rab$l_rbf], buffer+record_offset);
;  369          0602  2
;  370          0603  2 IF NULLPARAMETER(1)                     ! If FAB not specified,
;  371          0604  2 THEN
;  372          0605  2     put_line(bufdesc)                   ! Output the line
;  373          0606  2 ELSE
;  374          0607  2     put_line(bufdesc,.rab);             ! else output on specified rab
;  375          0608  2
;  376          0609  2 RETURN true;
;  377          0610  2
;  378          0611  1 END;
```

```
                                          003C 00000         .ENTRY  ECHO_RECORD, Save R2,R3,R4,R5                    ; 0561
                          5E      FF70  CE   9E 00002         MOVAB   -144(SP), SP
                 0000'    CF      0000G CF   D1 00007         CMPL    INPUT_LINENUM, LAST_ECHOED                       ; 0588
                                        53   13 0000E         BEQL    4$
                 0000'    CF      0000G CF   D0 00010         MOVL    INPUT_LINENUM, LAST_ECHOED                       ; 0591
     28            20             6E    00   2C 00017         MOVC5   #0, (SP), #32, #40, BUFFER                       ; 0593
                                    08  AE      0001C
                                    08  AE   9F 0001E         PUSHAB  BUFFER                                           ; 0594
                 0000V    CF            01   FB 00021         CALLS   #1, SET_LINE_NUMBER
                          50      0000G CF   3C 00026         MOVZWL  INPUT_RAB+34, R0                                 ; 0596
                 005D     8F            50   B1 0002B         CMPW    R0, #93
                                        04   1B 00030         BLEQU   1$
                          50        5D  8F   9A 00032         MOVZBL  #93, R0
                          6E        28  A0   9E 00036 1$:     MOVAB   40(R0), BUFDESC                                  ; 0598
                   04     AE        08  AE   9E 0003A         MOVAB   BUFFER, BUFDESC+4                                ; 0599
          30  AE 0000G    DF            50   28 0003F         MOVC3   INPUT_LENGTH, @INPUT_RAB+40, BUFFER+40           ; 0601
                                        6C   95 00046         TSTB    (AP)                                             ; 0603
                                        05   13 00048         BEQL    2$
                                    04  AC   D5 0004A         TSTL    4(AP)
                                        09   12 0004D         BNEQ    3$
                                        5E   DD 0004F 2$:     PUSHL   SP                                               ; 0605
                 0000V    CF            01   FB 00051         CALLS   #1, PUT_LINE
                                        0B   11 00056         BRB     4$
                                    04  AC   DD 00058 3$:     PUSHL   RAB                                              ; 0607
                                    04  AE   9F 0005B         PUSHAB  BUFDESC
                 0000V    CF            02   FB 0005E         CALLS   #2, PUT_LINE
                          50            01   D0 00063 4$:     MOVL    #1, R0                                           ; 0609
                                             04 00066         RET                                                      ; 0611

; Routine Size:  103 bytes,    Routine Base:  $CODE$ + 0184
```

```
;  380          0612  1 %SBTTL 'SET_LINE_NUMBER'
;  381          0613  1 ROUTINE set_line_number (buffer) =
;  382          0614  1
;  383          0615  1 !---
;  384          0616  1 !
;  385          0617  1 !       This routine increments the current line number
;  386          0618  1 !       and sets the line number into the output buffer
;  387          0619  1 !       at the correct position.
;  388          0620  1 !
;  389          0621  1 ! Inputs:
;  390          0622  1 !
;  391          0623  1 !       buffer = Address of buffer (line_size bytes)
;  392          0624  1 !
;  393          0625  1 ! Outputs:
;  394          0626  1 !
;  395          0627  1 !       None
;  396          0628  1 !---
;  397          0629  1
;  398          0630  2 BEGIN
;  399          0631  2
;  400          0632  2 MAP
;  401          0633  2     buffer:     REF VECTOR [,BYTE];     ! Address of buffer
;  402          0634  2
;  403          0635  2 LOCAL
;  404          0636  2     bufdesc:    VECTOR [2];             ! Descriptor of above string
;  405          0637  2
;  406          0638  2 bufdesc [0] = 4;                        ! Set length of buffer
;  407          0639  2 bufdesc [1] = .buffer + linenum_offset; ! Set address of buffer
;  408          0640  2
;  409          0641  2 $FAO(descriptor('!4UL'),0, bufdesc, .input_linenum);
;  410          0642  2
;  411          0643  2 RETURN true;
;  412          0644  2
;  413          0645  1 END;


                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                4C  55  34  21  0000C P.AAD:   .ASCII  \!4UL\                          ;
                                      00000004  00010 P.AAC:   .LONG   4                               ;
                                      00000000' 00014          .ADDRESS P.AAD                         ;


                                                .PSECT  $CODE$,NOWRT,2

                                          0000 00000 SET_LINE_NUMBER:
                                                     .WORD   Save nothing                             ; 0613
                                    5E            04  C2 00002          SUBL2   #4, SP                ;
                                                  04  DD 00005          PUSHL   #4                    ; 0638
                        04  AE          04  AC     23  C1 00007          ADDL3   #35, BUFFER, BUFDESC+4 ; 0639
                                          0000G   CF  DD 0000D          PUSHL   INPUT_LINENUM          ; 0641
                                            04    AE  9F 00011          PUSHAB  BUFDESC                ;
                                                  7E  D4 00014          CLRL    -(SP)                  ;
                                          0000'   CF  9F 00016          PUSHAB  P.AAC                  ;
                                00000000G   00        04  FB 0001A     CALLS   #4, SYS$FAO            ;
```

```
                                50          01  D0 00021          MOVL    #1, R0                          ; 0643
                                                04 00024          RET                                     ; 0645

; Routine Size:  37 bytes,    Routine Base:  $CODE$ + 01EB
```

```
;  415          0646  1 %SBTTL 'NEW_PAGE'
;  416          0647  1 GLOBAL ROUTINE new_page =
;  417          0648  1
;  418          0649  1 !---
;  419          0650  1 !
;  420          0651  1 !       This routine outputs the heading lines at the
;  421          0652  1 !       start of each page.  A page eject is issued here.
;  422          0653  1 !
;  423          0654  1 ! Inputs:
;  424          0655  1 !
;  425          0656  1 !       None
;  426          0657  1 !
;  427          0658  1 ! Outputs:
;  428          0659  1 !
;  429          0660  1 !       None
;  430          0661  1 !---
;  431          0662  1
;  432          0663  2 BEGIN
;  433          0664  2
;  434          0665  2 BIND
;  435          0666  2     input_nam = .input_fab [fab$l_nam]: BBLOCK,          ! Address of NAM block
;  436          0667  2     input_xabdat = .input_fab [fab$l_xab]: BBLOCK;       ! Address of XABDAT
;  437          0668  2
;  438          0669  2 LOCAL
;  439          0670  2     buffer:     VECTOR [line_size,BYTE],          ! Output buffer
;  440          0671  2     bufdesc:    VECTOR [2];                       ! Buffer descriptor
;  441          0672  2
;  442          0673  2 IF .page_number EQL 0                             ! If first page in output file,
;  443          0674  2 THEN
;  444          0675  2     lines_per_page = LIB$LP_LINES()-6;            ! then initialize lines per page
;  445          0676  2
;  446          0677  2 line_number = 0;                                  ! Reset line number to 0
;  447          0678  2 page_number = .page_number + 1;                   ! and increment page number
;  448          0679  2
;  449          0680  2 bufdesc [0] = line_size;                          ! Length of buffer
;  450          0681  2 bufdesc [1] = buffer;                             ! Address of buffer
;  451          0682  2
;  452        P 0683  2 $FAO(descriptor(
;  453        P 0684  2 '!^!31AS !28AS !20%D  VAX-11 Message V04-00                    Page!4UL'),
;  454          0685  2         bufdesc,bufdesc,module_name,title_text,0,.page_number);
;  455          0686  2
;  456          0687  2 put_line(bufdesc);                                 ! Output line
;  457          0688  2
;  458          0689  2 bufdesc [0] = line_size;                          ! Reset length of buffer
;  459          0690  2
;  460        P 0691  2 $FAO(descriptor('!61* !20%D  !44AD'),
;  461        P 0692  2         bufdesc,bufdesc,input_xabdat [xab$q_cdt],
;  462          0693  2         .input_nam [nam$b_rsl], .input_nam [nam$l_rsa]);
;  463          0694  2
;  464          0695  2 put_line(bufdesc);                                 ! Output line
;  465          0696  2
;  466          0697  2 bufdesc [0] = 1;                                  ! Reset to null line
;  467          0698  2 put_line(bufdesc);                                 ! Output a blank line
;  468          0699  2
;  469          0700  2 RETURN true;
;  470          0701  2
;  471          0702  1 END;
```

```
                                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

21 20 53 41 38 32 21 20 53 41 31 33 21 5E 21  00018 P.AAF:  .ASCII  \!^!31AS !28AS !20%D  VAX-11 Message V04-\
65 4D 20 31 31 2D 58 41 56 20 20 44 25 30 32  00027
                  2D 34 30 56 20 65 67 61 73  00036
20 20 20 20 20 20 20 20 20 20 20 20 20 20 30  00040          .ASCII  \00                    Page!4UL\
   4C 55 34 21 65 67 61 50 20 20 20 20 20 20  0004F
                                              0005D          .BLKB   3
                                  00000045    00060 P.AAE:   .LONG   69
                                  00000000'   00064          .ADDRESS P.AAF
34 34 21 20 20 44 25 30 32 21 20 2A 31 36 21  00068 P.AAH:   .ASCII  \!61* !20%D  !44AD\
                                        44 41  00077
                                              00079          .BLKB   3
                                  00000011    0007C P.AAG:   .LONG   17
                                  00000000'   00080          .ADDRESS P.AAH


                                                                                .PSECT  $CODE$,NOWRT,2

                                     007C 00000          .ENTRY  NEW_PAGE, Save R2,R3,R4,R5,R6                    ; 0647
                              56  0000V  CF   9E 00002    MOVAB   PUT_LINE, R6                                     ;
                              55 00000000G 00  9E 00007    MOVAB   SYS$FAO, R5                                      ;
                              54  0000'  CF   9E 0000E    MOVAB   PAGE_NUMBER, R4                                  ;
                              5E  FF70   CE   9E 00013    MOVAB   -144(SP), SP                                     ;
                              52  0000G  CF   D0 00018    MOVL    INPUT_FAB+40, R2                                 ; 0666
                              53  0000G  CF   D0 0001D    MOVL    INPUT_FAB+36, R3                                 ; 0667
                                  64     D5   00022      TSTL    PAGE_NUMBER                                      ; 0673
                                  0C     12   00024      BNEQ    1$                                               ;
                00000000G     00          00   FB 00026 CALLS   #0, LIB$LP_LINES                                 ; 0675
                      08      A4    FA    A0   9E 0002D MOVAB   -6(R0), LINES_PER_PAGE                           ;
                              04    A4    D4   00032 1$: CLRL   LINE_NUMBER                                      ; 0677
                                  64     D6   00035      INCL    PAGE_NUMBER                                      ; 0678
                          6E      85    8F   9A 00037    MOVZBL  #133, BUFDESC                                    ; 0680
                      04  AE      08    AE   9E 0003B    MOVAB   BUFFER, BUFDESC+4                                ; 0681
                                  64     DD   00040      PUSHL   PAGE_NUMBER                                      ; 0685
                                  7E     D4   00042      CLRL    -(SP)                                            ;
                                0000G    CF   9F 00044    PUSHAB  TITLE_TEXT                                       ;
                                0000G    CF   9F 00048    PUSHAB  MODULE_NAME                                      ;
                                  10     AE   9F 0004C    PUSHAB  BUFDESC                                          ;
                                  14     AE   9F 0004F    PUSHAB  BUFDESC                                          ;
                                0000'    CF   9F 00052    PUSHAB  P.AAE                                            ;
                              65                  07   FB 00056 CALLS   #7, SYS$FAO                              ;
                                                  5E   DD 00059 PUSHL   SP                                       ; 0687
                              66                  01   FB 0005B CALLS   #1, PUT_LINE                             ;
                              6E      85    8F   9A 0005E    MOVZBL  #133, BUFDESC                                    ; 0689
                              7E      04    A2   DD 00062    PUSHL   4(R2)                                            ; 0693
                                      03    A2   9A 00065    MOVZBL  3(R2), -(SP)                                     ;
                                      14    A3   9F 00069    PUSHAB  20(R3)                                           ;
                                      0C    AE   9F 0006C    PUSHAB  BUFDESC                                          ;
                                      10    AE   9F 0006F    PUSHAB  BUFDESC                                          ;
                                    0000'   CF   9F 00072    PUSHAB  P.AAG                                            ;
                              65                  06   FB 00076 CALLS   #6, SYS$FAO                              ;
                                                  5E   DD 00079 PUSHL   SP                                       ; 0695
```

```
                66              01  FB 0007B         CALLS   #1, PUT_LINE                    ;
                6E              01  D0 0007E         MOVL    #1, BUFDESC                     ; 0697
                                5E  DD 00081         PUSHL   SP                              ; 0698
                66              01  FB 00083         CALLS   #1, PUT_LINE                    ;
                50              01  D0 00086         MOVL    #1, R0                          ; 0700
                                    04 00089         RET                                     ; 0702

; Routine Size:  138 bytes,    Routine Base:  $CODE$ + 0210
```

```
;  473     0703  1 %SBTTL 'PUT_LINE'
;  474     0704  1 ROUTINE put_line (line_desc, rab) =
;  475     0705  1
;  476     0706  1 !---
;  477     0707  1 !
;  478     0708  1 !       This routine outputs a given record to the listing
;  479     0709  1 !       file.
;  480     0710  1 !
;  481     0711  1 ! Inputs:
;  482     0712  1 !
;  483     0713  1 !       line_desc = Address of record descriptor
;  484     0714  1 !       rab = Address of output rab to use (if not specified, use listing_rab)
;  485     0715  1 !
;  486     0716  1 ! Outputs:
;  487     0717  1 !
;  488     0718  1 !       None, errors are signaled
;  489     0719  1 !---
;  490     0720  1
;  491     0721  2 BEGIN
;  492     0722  2
;  493     0723  2 BUILTIN
;  494     0724  2     NULLPARAMETER;                  ! True if not specified
;  495     0725  2
;  496     0726  2 MAP
;  497     0727  2     line_desc:  REF VECTOR;         ! Address of a descriptor
;  498     0728  2
;  499     0729  2 LOCAL
;  500     0730  2     outfab:     REF BBLOCK,         ! Output FAB
;  501     0731  2     outrab:     REF BBLOCK,         ! Output RAB
;  502     0732  2     status;                         ! Status code
;  503     0733  2
;  504     0734  2 IF NULLPARAMETER(2)                 ! If RAB not specified,
;  505     0735  2 THEN
;  506     0736  2     outrab = listing_rab            ! then use listing file rab
;  507     0737  2 ELSE
;  508     0738  2     outrab = .rab;                  ! else, use RAB address
;  509     0739  2 outfab = .outrab [rab$l_fab];       ! and get FAB address
;  510     0740  2
;  511     0741  2 IF .outfab [fab$w_ifi] EQL 0        ! If file not opened
;  512     0742  2 THEN
;  513     0743  2     RETURN true;                    ! then do nothing
;  514     0744  2
;  515     0745  2 line_number = .line_number + 1;     ! Increment line number
;  516     0746  2
;  517     0747  2 IF .line_number GEQ .lines_per_page ! If exceeded lines per page,
;  518     0748  2 THEN
;  519     0749  2     new_page();                     ! then cause page eject
;  520     0750  2
;  521     0751  2 outrab [rab$w_rsz] = .line_desc [0];    ! Set length of record
;  522     0752  2 outrab [rab$l_rbf] = .line_desc [1];    ! Set address of record
;  523     0753  2
;  524     0754  2 status = $PUT(RAB=.outrab);         ! Output line
;  525     0755  2
;  526     0756  2 IF NOT .status                      ! If error detected,
;  527     0757  2 THEN
;  528     0758  2     rms_error(emsg(writeerr),.outfab,.outrab);  ! report error
;  529     0759  2
```

```
;  530          0760  2 RETURN .status;                    ! return with status
;  531          0761  2
;  532          0762  1 END;


                                                .EXTRN  SYS$PUT

                                        001C 00000 PUT_LINE:
                                                .WORD   Save R2,R3,R4                   ; 0704
                            02                  6C  91 00002        CMPB    (AP), #2    ; 0734
                                                05  1F 00005        BLSSU   1$          ;
                                          08    AC  D5 00007        TSTL    8(AP)       ;
                                                07  12 0000A        BNEQ    2$          ;
                            52          0000G   CF  9E 0000C 1$:    MOVAB   LISTING_RAB, OUTRAB     ; 0736
                                                04  11 00011        BRB     3$          ;
                            52            08    AC  D0 00013 2$:    MOVL    RAB, OUTRAB             ; 0738
                            54            3C    A2  D0 00017 3$:    MOVL    60(OUTRAB), OUTFAB      ; 0739
                                          02    A4  B5 0001B        TSTW    2(OUTFAB)               ; 0741
                                                04  12 0001E        BNEQ    4$          ;
                            50                  01  D0 00020        MOVL    #1, R0                  ; 0743
                                                04  00023        RET                     ;
                                        0000'   CF  D6 00024 4$:    INCL    LINE_NUMBER             ; 0745
                      0000'  CF         0000'   CF  D1 00028        CMPL    LINE_NUMBER, LINES_PER_PAGE     ; 0747
                                                05  19 0002F        BLSS    5$          ;
                        FF40  CF                00  FB 00031        CALLS   #0, NEW_PAGE            ; 0749
                            50            04    AC  D0 00036 5$:    MOVL    LINE_DESC, R0           ; 0751
                          22  A2                60  B0 0003A        MOVW    (R0), 34(OUTRAB)        ;
                          28  A2            04  A0  D0 0003E        MOVL    4(R0), 40(OUTRAB)       ; 0752
                                                52  DD 00043        PUSHL   OUTRAB                  ; 0754
                 00000000G  00                  01  FB 00045        CALLS   #1, SYS$PUT             ;
                            53                  50  D0 0004C        MOVL    R0, STATUS              ;
                            0F                  53  E8 0004F        BLBS    STATUS, 6$              ; 0756
                                                52  DD 00052        PUSHL   OUTRAB                  ; 0758
                                                54  DD 00054        PUSHL   OUTFAB                  ;
                                    009710D2    8F  DD 00056        PUSHL   #9900242                ;
                                0000G   CF              03  FB 0005C        CALLS   #3, RMS_ERROR   ;
                                        50              53  D0 00061 6$:    MOVL    STATUS, R0      ; 0760
                                                        04  00064        RET                     ; 0762

; Routine Size:  101 bytes,    Routine Base:  $CODE$ + 029A
```

```
;  534          0763  1 %SBTTL 'ERROR_SUMMARY'
;  535          0764  1 GLOBAL ROUTINE error_summary (rab) =
;  536          0765  1
;  537          0766  1 !---
;  538          0767  1 !
;  539          0768  1 !       This routine outputs a summary of the errors detected.
;  540          0769  1 !
;  541          0770  1 ! Inputs:
;  542          0771  1 !
;  543          0772  1 !       rab = Address of RAB to use
;  544          0773  1 !
;  545          0774  1 ! Outputs:
;  546          0775  1 !
;  547          0776  1 !       None
;  548          0777  1 !---
;  549          0778  1
;  550          0779  2 BEGIN
;  551          0780  2
;  552          0781  2 LOCAL
;  553          0782  2     bufdesc:    VECTOR [2];             ! Buffer descriptor
;  554          0783  2
;  555          0784  2 bufdesc [0] = 0;                        ! Null line
;  556          0785  2 put_line(bufdesc, .rab);                ! Output a blank line
;  557          0786  2
;  558          0787  2 bufdesc [0] = line_size;
;  559        P 0788  2 $FAO(descriptor('There were !UL error!%S, !UL warning!%S, and !UL informational message!%S issued.'),
;  560          0789  2         bufdesc,bufdesc,.errors,.warnings,.infos);
;  561          0790  2
;  562          0791  2 put_line(bufdesc, .rab);                ! Output error summary
;  563          0792  2
;  564          0793  2 RETURN true;
;  565          0794  2
;  566          0795  1 END;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

20 4C 55 21 20 65 72 65 77 20 65 72 65 68 54  00084 P.AAJ:  .ASCII  \There were !UL error!%S, !UL warning!%S,\  ;
77 20 4C 55 21 20 2C 53 25 21 72 6F 72 72 65  00093                                                          ;
                  2C 53 25 21 67 6E 69 6E 72  000A2                                                          ;
6D 72 6F 66 6E 69 20 4C 55 21 20 64 6E 61 20  000AC          .ASCII  \ and !UL informational message!%S issued\ ;
65 67 61 73 73 65 6D 20 6C 61 6E 6F 69 74 61  000BB                                                          ;
                  64 65 75 73 73 69 20 53 25  000CA                                                          ;
                                     2E       000D4          .ASCII  \.\                                    ;
                                              000D5          .BLKB   3
                                 00000051     000D8 P.AAI:   .LONG   81                                     ;
                                 00000000'    000DC          .ADDRESS P.AAJ                                 ;


                                                        .PSECT  $CODE$,NOWRT,2

                                    0000 00000          .ENTRY  ERROR_SUMMARY, Save nothing                ; 0764
                              5E       04   C2 00002    SUBL2   #4, SP                                     ;
                                       7E   D4 00005    CLRL    BUFDESC                                    ; 0784
                                  04   AC   DD 00007    PUSHL   RAB                                        ; 0785
```

```
                                 04   AE  9F 0000A         PUSHAB  BUFDESC                                ;
                              8A AF       02  FB 0000D         CALLS   #2, PUT_LINE                    ;
                              6E 85       8F  9A 00011         MOVZBL  #133, BUFDESC                   ; 0787
                              7E 0000'    CF  7D 00015         MOVQ    WARNINGS, -(SP)                 ; 0789
                                 0000'    CF  DD 0001A         PUSHL   ERRORS                          ;
                                 0C       AE  9F 0001E         PUSHAB  BUFDESC                         ;
                                 10       AE  9F 00021         PUSHAB  BUFDESC                         ;
                                 0000'    CF  9F 00024         PUSHAB  P.AAI                           ;
                        00000000G 00          06  FB 00028         CALLS   #6, SYS$FAO                     ;
                                 04       AC  DD 0002F         PUSHL   RAB                             ; 0791
                                 04       AE  9F 00032         PUSHAB  BUFDESC                         ;
                              FF61 CF          02  FB 00035         CALLS   #2, PUT_LINE                    ;
                              50               01  D0 0003A         MOVL    #1, R0                          ; 0793
                                               04  0003D         RET                                     ; 0795

; Routine Size:  62 bytes,    Routine Base:  $CODE$ + 02FF
```

```
;  568          0796  1 %SBTTL 'END_LISTING'
;  569          0797  1 GLOBAL ROUTINE end_listing (rab) =
;  570          0798  1
;  571          0799  1 !---
;  572          0800  1 !
;  573          0801  1 !       This routine is called to output the compile run
;  574          0802  1 !       statistics collected during execution.
;  575          0803  1 !
;  576          0804  1 ! Inputs:
;  577          0805  1 !
;  578          0806  1 !       rab = Address of RAB to use
;  579          0807  1 !
;  580          0808  1 ! Outputs:
;  581          0809  1 !
;  582          0810  1 !       None
;  583          0811  1 !---
;  584          0812  1
;  585          0813  2 BEGIN
;  586          0814  2 BUILTIN
;  587          0815  2     ACTUALCOUNT;
;  588          0816  2
;  589          0817  2 error_summary(.rab);                    ! Output error summary
;  590          0818  2
;  591          0819  2 IF ACTUALCOUNT() EQL 0 THEN
;  592          0820  2     put_line (command_line_desc)
;  593          0821  2 ELSE
;  594          0822  2     IF .rab EQLA listing_rab THEN
;  595          0823  2         put_line (command_line_desc);
;  596          0824  2
;  597          0825  2
;  598          0826  2 RETURN true;
;  599          0827  2
;  600          0828  1 END;


                                    0000 00000          .ENTRY  END_LISTING, Save nothing                ; 0797
                                04  AC  DD 00002          PUSHL   RAB                                      ; 0817
                            B9  AF      01  FB 00005          CALLS   #1, ERROR_SUMMARY
                                        6C  95 00009          TSTB    (AP)                                     ; 0819
                                        0B  13 0000B          BEQL    1$
                                50      CF  9E 0000D          MOVAB   LISTING_RAB, R0                          ; 0822
                                50  04  AC  D1 00012          CMPL    RAB, R0
                                        09  12 00016          BNEQ    2$
                                    0000G   CF  9F 00018 1$:      PUSHAB  COMMAND_LINE_DESC                    ; 0823
                            FF3C    CF      01  FB 0001C          CALLS   #1, PUT_LINE
                                50          01  D0 00021 2$:      MOVL    #1, R0                               ; 0826
                                            04  00024          RET                                          ; 0828

; Routine Size:  37 bytes,    Routine Base:  $CODE$ + 033D
```

```
;  602          0829  1 %SBTTL ''
;  603          0830  1 END
;  604          0831  0 ELUDOM



;                                   PSECT SUMMARY
;
;       Name                    Bytes                           Attributes
;
;  $OWN$                          28  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                        866  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $PLIT$                        224  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)



;                                 Library Statistics
;                                          -------- Symbols --------      Pages       Processing
;       File                               Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1        9776        35         0         581       00:01.0
;  _$255$DUA28:[SYSLIB]TPAMAC.L32;1           42         0         0          14       00:00.1



;                                    COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:LISTING/OBJ=OBJ$:LISTING MSRC$:LISTING/UPDATE=(ENH$:LISTING)

;  Size:          866 code + 252 data bytes
;  Run Time:          00:17.1
;  Elapsed Time:      00:55.6
;  Lines/CPU Min:      2919
;  Lexemes/CPU-Min:  21498
;  Memory Used:  117 pages
;  Compilation Complete
```

MSGINT
MDL

SDLGENREQ
REQ

MSGDEF
SDL

MAIN
LIS

CVTMSGCOM
FOR

LISTING
LIS

CVTMSG
LIS

MSG
REQ