```
MMM         MMM     SSSSSSSSSSSS     GGGGGGGGGGGG  FFFFFFFFFFFFFFF     IIIIIIIII     LLL
MMM         MMM     SSSSSSSSSSSS     GGGGGGGGGGGG  FFFFFFFFFFFFFFF     IIIIIIIII     LLL
MMM         MMM     SSSSSSSSSSSS     GGGGGGGGGGGG  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              FFFFFFFFFFFF           III        LLL
MMM         MMM     SSSSSSSSS     GGG              FFFFFFFFFFFF           III        LLL
MMM         MMM     SSSSSSSSS     GGG              FFFFFFFFFFFF           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  SSSSSSSSSSSS        GGGGGGGGG     FFF                 IIIIIIIII     LLLLLLLLLLLLLLL
MMM         MMM  SSSSSSSSSSSS        GGGGGGGGG     FFF                 IIIIIIIII     LLLLLLLLLLLLLLL
MMM         MMM  SSSSSSSSSSSS        GGGGGGGGG     FFF                 IIIIIIIII     LLLLLLLLLLLLLLL
```

FILEID**PARSE

D 8

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


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

```
0 MODULE parse (%TITLE 'PARSE THE MESSAGE SOURCE FILE' IDENT = 'V04-000') =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1
1 !++
1 ! FACILITY:  Message compiler
1 !
1 ! ABSTRACT:
1 !
1 !       This compiler translated message definition language
1 !       into object modules
1 !
1 ! ENVIRONMENT:
1 !
1 !       VAX/VMS operating system. unprivileged user mode,
1 !
1 ! AUTHOR:  Tim Halvorsen, Nov 1979
1 !
1 ! Modified by:
1 !
1 !       V03-003 GJA0063         Greg Awdziewicz         17-Jan-1984
1 !               - Avoid upcasing the titles in the listing output by the
1 !               message compiler.
1 !               - Add source title and subtitles in this module.
1 !
1 !       002     JWT0048         Jim Teague      05-Aug-1982
1 !               Touch up SDL output.
1 !
1 !       001     JWT0025         Jim Teague      17-Mar-1982
1 !               Add the / delimiter for .IDENT fields
1 !
1 !
1 !--
1
```

```
      0058  1 !
      0059  1 ! Include files
      0060  1 !
      0061  1
      0062  1 LIBRARY 'SYS$LIBRARY:STARLET';         ! VAX/VMS common definitions
      0063  1
      0064  1 LIBRARY 'SYS$LIBRARY:TPAMAC';          ! TPARSE definitions
      0065  1
      0066  1 REQUIRE 'SRC$:MSG.REQ';                ! Command definitions
```

```
1
1 !
1 ! Table of contents
1 !
1
1 FORWARD ROUTINE
1     parse_file,                           ! Parse input file
1     get_record,                           ! Get next input record
1     message_init,                         ! Initialize for next message
1     message_defn,                         ! Process a message definition
1     add_message,                          ! Add message to msg defn list
1     facility_init,                        ! Initialize for next facility
1     facility_defn,                        ! Process a facility definition
1     add_facility,                         ! Add facility to fac defn list
1     add_symbol,                           ! Add symbol to symbol table
1     lookup_symbol,                        ! Lookup symbol in symbol table
1     find_eos,                             ! Find end of message string
1     find_endvers,                         ! Store delimited ident/version string
1     get_cont_line,                        ! Get continuation line and plug in
1     define_literal,                       ! Define a user literal
1     set_title,                            ! Store title string
1     set_module,                           ! Store module name string
1     build_version,                        ! Store undelimited version/ident string
1     store_number,                         ! Store/check numeric qualifier value
1     store_string,                         ! Store/check string qualifier value
1     allocate,                             ! Allocate dynamic storage
1     deallocate,                           ! Deallocate dynamic storage
1     comment;                              ! Call MDL or SDL to output comment
1
1 !
1 ! Storage definitions
1 !
1
1 LITERAL
1     form_feed = 12,                       ! Form feed character
1     facility_bufsiz = 9,                  ! Maximum size of facility name
1     prefix_bufsiz = obj$c_symsiz,         ! Max size of facility prefix
1     defpre_bufsiz = 9,                    ! Maximum size of /PREFIX string
1     macro_bufsiz = 15,                    ! Maximum size of facility macro name
1     symbol_bufsiz = obj$c_symsiz,         ! Maximum size of symbol name
1     sym_plus_pre = obj$c_symsiz,          ! Maximum size of symbol + prefix
1     ident_bufsiz = 15,                    ! Maximum size of /IDENT string
1     message_bufsiz = 256,                 ! Maximum size of message text string
1     title_bufsiz = 128;                   ! Maximum size of title string
1
1 GLOBAL
1     facility_buffer:    BBLOCK [facility_bufsiz], ! Facility name buffer
1     facility_name:      VECTOR [2]        ! Descriptor of facility name
1                         INITIAL(0,facility_buffer),
1     message_header:     INITIAL(0),       ! Listhead for CODE blocks
1     facility_header:    INITIAL(0),       ! Listhead for FAC blocks
1     symbol_header:      INITIAL(0),       ! Listhead for symbol table list
1     num_messages:       INITIAL(0),       ! Number of messages defined
1     msg_space:          INITIAL(0),       ! Total space used by MSG blocks
1     num_facilities:     INITIAL(0),       ! Number of facilities defined
1     fac_space:          INITIAL(2),       ! Total space needed for facility table
1                                           ! initially 2 bytes (list terminator)
```

```
1       num_files:          INITIAL(0),      ! Total files acceped as input
1       title_buffer:       BBLOCK[title_bufsiz],  ! Title text buffer
1       title_text:         VECTOR[2]        ! Module title text
1                           INITIAL(0,title_buffer),
1       input_record:       VECTOR[2],       ! Input record descriptor
1       input_linenum,                       ! Line number of input record
1       version_buffer:     BBLOCK[obj$c_symsiz],  ! Version/ident string buffer
1       version_num :       VECTOR[2]        ! Descriptor for version/ident
1                           INITIAL (0,version_buffer);
1
1 OWN
1       tparse_block: BBLOCK[tpa$c_length0] ! TPARSE parameter block
1                   PRESET( [tpa$l_count]   = tpa$k_count0,
1                           [tpa$l_options] = tpa$m_abbrev),
1       facility_number,                     ! Current facility number
1       facility_flags:     BITVECTOR [32],  ! Flags describing current facility
1       defpre_buffer:      BBLOCK [defpre_bufsiz], ! Default prefix buffer
1       prefix_buffer:      BBLOCK [prefix_bufsiz], ! Prefix buffer
1       default_prefix:     VECTOR [2]       ! Symbol prefix for current facility
1                           INITIAL(0,defpre_buffer),
1       default_sev,                         ! Default severity
1       default_lang,                        ! Default language ident number
1       macro_buffer:       BBLOCK [macro_bufsiz],  ! Macro name buffer
1       macro_name:         VECTOR [2]       ! MDL macro name for facility
1                           INITIAL(0,macro_buffer),
1       message_number,                      ! Current message number
1       symbol_buffer:      BBLOCK [symbol_bufsiz], ! Symbol name buffer
1       symbol_name:        VECTOR [2]       ! Symbol name descriptor
1                           INITIAL(0,symbol_buffer),
1       severity_value,                      ! Severity for message
1       faocnt_value,                        ! FAOCNT value
1       ident_buffer:       BBLOCK [ident_bufsiz],  ! IDENT string buffer
1       ident_value:        VECTOR [2]       ! IDENT descriptor
1                           INITIAL(0,ident_buffer),
1       detail_value,                        ! DETAIL value
1       lang_value,                          ! LANG numeric value (see $MSGDEF)
1       userval_value,                       ! USERVAL value
1       message_buffer:     BBLOCK [message_bufsiz], ! Message text buffer
1       message_text:       VECTOR [2]       ! Message text descriptor
1                           INITIAL(0,message_buffer),
1       module_buffer:      BBLOCK [obj$c_symsiz],  ! Module name string buffer
1       literal_name:       VECTOR [2],      ! Descriptor of literal symbol name
1       literal_value,                       ! Value to be assigned to literal
1
1       line_output:        BYTE,            ! True if line was output
1       new_line:           INITIAL(true);   ! True if new line should be started for comment
1
1 LITERAL
1       shared_bit = 0,                      ! /SHARED bit number
1       shared_mask = 1,                     ! /SHARED mask
1       system_bit = 1,                      ! /SYSTEM bit number
1       system_mask = 2,                     ! /SYSTEM mask
1       prefix_bit = 2,                      ! /PREFIX bit number
1       prefix_mask = 4,                     ! /PREFIX mask
1       macro_mask = 8,                      ! /MACRO mask
1       literal_flag = 0;                    ! Indicate literal call to mdlgen or sdlgen
1
```

```
;  182          0418  1 !
;  183          0419  1 ! External storage
;  184          0420  1 !
;  185          0421  1
;  186          0422  1 EXTERNAL
;  187          0423  1     cli_flags:          BITVECTOR,       ! CLI qualifier bitmap
;  188          0424  1     module_name:        VECTOR,          ! Module name descriptor
;  189          0425  1     input_fab:          BBLOCK,          ! Input file FAB
;  190          0426  1     input_rab:          BBLOCK;          ! Input file RAB
;  191          0427  1
;  192          0428  1 !
;  193          0429  1 ! External routines
;  194          0430  1 !
;  195          0431  1
;  196          0432  1 EXTERNAL ROUTINE
;  197          0433  1     line_with_value,                     ! Output a line with a hex value
;  198          0434  1     echo_record,                         ! Output a line w/only record
;  199          0435  1     new_page,                            ! Cause a page eject
;  200          0436  1     syntax_error,                        ! Report syntax error
;  201          0437  1     lib$tparse: ADDRESSING_MODE(GENERAL),       ! Parsing routines
;  202          0438  1     lib$get_vm: ADDRESSING_MODE(GENERAL),       ! Allocate dynamic storage
;  203          0439  1     lib$free_vm: ADDRESSING_MODE(GENERAL),      ! Deallocate dynamic storage
;  204          0440  1     rms_error,                           ! Signal RMS-type error
;  205          0441  1     mdl_start_struc,                     ! Start structure definition
;  206          0442  1     mdl_define_constant,                 ! Define message or literal constant
;  207          0443  1     mdl_end_struc,                       ! End structure definition
;  208          0444  1     mdl_comment,                         ! Output a comment
;  209          0445  1     mdl_put_record,                      ! Output a record
;  210          0446  1
;  211          0447  1     sdl_start_mod,                       ! Start SDL module definition
;  212          0448  1     sdl_define_constant,                 ! Define message or literal constant
;  213          0449  1     sdl_end_mod,                         ! End SDL module definition
;  214          0450  1     sdl_comment,                         ! Output a comment
;  215          0451  1     sdl_put_record;                      ! Output a record
;  216          0452  1
;  217          0453  1 ROUTINE null: NOVALUE =;

                                                        .TITLE  PARSE PARSE THE MESSAGE SOURCE FILE
                                                        .IDENT  \V04-000\

                                                        .PSECT  $OWN$,NOEXE,2

                          00000002  00000008  00000 TPARSE_BLOCK:
                                                        .LONG   8, 2                                    ;
                                              00008     .BLKB   28
                                              00024 FACILITY_NUMBER:
                                                        .BLKB   4
                                              00028 FACILITY_FLAGS:
                                                        .BLKB   4
                                              0002C DEFPRE_BUFFER:
                                                        .BLKB   9
                                              00035     .BLKB   3
                                              00038 PREFIX_BUFFER:
                                                        .BLKB   31
                                              00057     .BLKB   1
                                    00000000  00058 DEFAULT_PREFIX:
```

```
                                  .LONG   0
00000000'  0005C                  .ADDRESS DEFPRE_BUFFER                      ;
           00060 DEFAULT_SEV:
                                  .BLKB   4
           00064 DEFAULT_LANG:
                                  .BLKB   4
           00068 MACRO_BUFFER:
                                  .BLKB   15
           00077                  .BLKB   1
00000000   00078 MACRO_NAME:
                                  .LONG   0
00000000'  0007C                  .ADDRESS MACRO_BUFFER                       ;
           00080 MESSAGE_NUMBER:
                                  .BLKB   4
           00084 SYMBOL_BUFFER:
                                  .BLKB   31
           000A3                  .BLKB   1
00000000   000A4 SYMBOL_NAME:
                                  .LONG   0
00000000'  000A8                  .ADDRESS SYMBOL_BUFFER                      ;
           000AC SEVERITY_VALUE:
                                  .BLKB   4
           000B0 FAOCNT_VALUE:
                                  .BLKB   4
           000B4 IDENT_BUFFER:
                                  .BLKB   15
           000C3                  .BLKB   1
00000000   000C4 IDENT_VALUE:
                                  .LONG   0
00000000'  000C8                  .ADDRESS IDENT_BUFFER                       ;
           000CC DETAIL_VALUE:
                                  .BLKB   4
           000D0 LANG_VALUE:
                                  .BLKB   4
           000D4 USERVAL_VALUE:
                                  .BLKB   4
           000D8 MESSAGE_BUFFER:
                                  .BLKB   256
00000000   001D8 MESSAGE_TEXT:
                                  .LONG   0
00000000'  001DC                  .ADDRESS MESSAGE_BUFFER                     ;
           001E0 MODULE_BUFFER:
                                  .BLKB   31
           001FF                  .BLKB   1
           00200 LITERAL_NAME:
                                  .BLKB   8
           00208 LITERAL_VALUE:
                                  .BLKB   4
           0020C LINE_OUTPUT:
                                  .BLKB   1
           0020D                  .BLKB   3
00000001   00210 NEW_LINE:
                                  .LONG   1                                   ;

                                  .PSECT  $GLOBAL$,NOEXE,2

           00000 FACILITY_BUFFER::
```

```
                                .BLKB   9
                     00009      .BLKB   3
          00000000   0000C FACILITY_NAME::
                                .LONG   0                                   ;
          00000000'  00010      .ADDRESS FACILITY_BUFFER                    ;
          00000000   00014 MESSAGE_HEADER::
                                .LONG   0                                   ;
          00000000   00018 FACILITY_HEADER::
                                .LONG   0                                   ;
          00000000   0001C SYMBOL_HEADER::
                                .LONG   0                                   ;
          00000000   00020 NUM_MESSAGES::
                                .LONG   0                                   ;
          00000000   00024 MSG_SPACE::
                                .LONG   0                                   ;
          00000000   00028 NUM_FACILITIES::
                                .LONG   0                                   ;
          00000002   0002C FAC_SPACE::
                                .LONG   2                                   ;
          00000000   00030 NUM_FILES::
                                .LONG   0                                   ;
                     00034 TITLE_BUFFER::
                                .BLKB   128
          00000000   000B4 TITLE_TEXT::
                                .LONG   0                                   ;
          00000000'  000B8      .ADDRESS TITLE_BUFFER                       ;
                     000BC INPUT_RECORD::
                                .BLKB   8
                     000C4 INPUT_LINENUM::
                                .BLKB   4
                     000C8 VERSION_BUFFER::
                                .BLKB   31
                     000E7      .BLKB   1
          00000000   000E8 VERSION_NUM::
                                .LONG   0                                   ;
          00000000'  000EC      .ADDRESS VERSION_BUFFER                     ;

                                .EXTRN  CLI_FLAGS, MODULE_NAME
                                .EXTRN  INPUT_FAB, INPUT_RAB
                                .EXTRN  LINE_WITH_VALUE
                                .EXTRN  ECHO_RECORD, NEW_PAGE
                                .EXTRN  SYNTAX_ERROR, LIB$TPARSE
                                .EXTRN  LIB$GET_VM, LIB$FREE_VM
                                .EXTRN  RMS_ERROR, MDL_START_STRUC
                                .EXTRN  MDL_DEFINE_CONSTANT
                                .EXTRN  MDL_END_STRUC, MDL_COMMENT
                                .EXTRN  MDL_PUT_RECORD, SDL_START_MOD
                                .EXTRN  SDL_DEFINE_CONSTANT
                                .EXTRN  SDL_END_MOD, SDL_COMMENT
                                .EXTRN  SDL_PUT_RECORD

                                .PSECT  $CODE$,NOWRT,2

                0000 00000 NULL:   .WORD   Save nothing                     ; 0453
                  04 00002         RET                                      ;

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

```
;  219          0454  1
;  220          0455  1 !
;  221          0456  1 !       Message definition language parse tables
;  222          0457  1 !
;  223          0458  1
;  224          0459  1 MACRO ap_setup = BUILTIN AP; MAP ap: REF BBLOCK%;
;  225          0460  1 ROUTINE set_number = (ap_setup; ap [tpa$l_number] = .ap [tpa$l_param]; true);



                                        0000 00000 SET_NUMBER:
                                                         .WORD   Save nothing                                 ; 0460
                    1C   AC        20   AC  D0 00002     MOVL    32(AP), 28(AP)                               ;
                         50             01  D0 00007     MOVL    #1, R0                                       ;
                                            04 0000A     RET                                                  ;

; Routine Size:  11 bytes,    Routine Base:  $CODE$ + 0003


;  226          0461  2 ROUTINE find_bracket = (ap_setup; BUILTIN CALLG; ap [tpa$l_char] = '>';
;  227          0462  1                          callg(.ap, find_eos));



                                        0000 00000 FIND_BRACKET:
                                                         .WORD   Save nothing                                 ; 0461
                    18   AC             3E  D0 00002     MOVL    #62, 24(AP)                                  ;
                 0000V   CF             6C  FA 00006     CALLG   (AP), FIND_EOS                               ; 0462
                                            04 0000B     RET                                                  ;

; Routine Size:  12 bytes,    Routine Base:  $CODE$ + 000E


;  228          0463  1 FORWARD ROUTINE init_stack;
;  229          0464  1
;  230          0465  1 $init_state(parse_states,parse_keys);   ! Define start of parse table
;  231          0466  1
;  232          0467  1 !
;  233          0468  1 !       Dispatch to the various command parsers
;  234          0469  1 !
;  235          0470  1
;  236        P 0471  1 $state(,
;  237          0472  1         (tpa$_lambda,,init_stack));     ! Init the stack each time thru
;  238          0473  1
;  239        P 0474  1 $state(,
;  240          0475  1         (tpa$_lambda,,message_init));   ! Init message cells each time thru
;  241          0476  1
;  242        P 0477  1 $state(main,
;  243        P 0478  1         ((contin),main,get_cont_line),
;  244        P 0479  1         ('!',tpa$_exit,comment),
;  245        P 0480  1         (tpa$_eos,tpa$_exit),
;  246        P 0481  1         ('.',directive),
;  247        P 0482  1         (tpa$_symbol,definition,store_string,,,
    1                   PLIT(symbol_name,1,symbol_bufsiz)));
    1
P   1 $state(directive,
P   1         ('FACILITY',facility,facility_init),
P   1         ('SEVERITY',severity),
P   1         ('LANGUAGE',language),
P   1         ('IDENT',ident1),
P   1         ('BASE',base),
P   1         ('LITERAL',literal_stmt),
P   1         ('PAGE',end_line,new_page),
P   1         ('TITLE',title),
    1         ('END',end_line,facility_init));
    1
    1 !
    1 !       Check for continuation line (trailing dash followed by eos or comment)
    1 !
    1
P   1 $state(contin,
    1         ('-'));
P   1 $state(,
P   1         ('!',tpa$_exit,comment),
    1         (tpa$_eos,tpa$_exit));
    1
    1 !
    1 !       Come here after command processed - check for end of string or comment
    1 !
    1
P   1 $state(end_line,
P   1         ((contin),end_line,get_cont_line),
P   1         ('!',tpa$_exit,comment),
    1         (tpa$_eos,tpa$_exit));
    1
    1 !
    1 !       Process FACILITY command to set facility name and number
    1 !
    1
P   1 $state(facility,
P   1         ((contin),facility,get_cont_line),
P   1         ('/'),
    1         (tpa$_lambda,fac10));
    1
P   1 $state(,
    1         ((facil_qual),facility));
    1
P   1 $state(fac10,
    1         (tpa$_symbol,,store_string,,,PLIT(facility_name,1,facility_bufsiz)));
    1
P   1 $state(fac15,
P   1         ((contin),fac15,get_cont_line),
P   1         (','),
    1         (tpa$_lambda));
    1
P   1 $state(fac18,
P   1         ((contin),fac18,get_cont_line),
P   1         ((expression),,store_number,,,
    1                 PLIT(facility_number,0,1^($FIELDWIDTH(sts$v_fac_no)-1)-1)));
    1
```

```
;  305        P 0540  1 $state(fac20,
;  306        P 0541  1           ((contin),fac20,get_cont_line),
;  307        P 0542  1           ('/'),
;  308          0543  1           (tpa$_lambda,end_line,facility_defn));
;  309          0544  1
;  310        P 0545  1 $state(,
;  311          0546  1           ((facil_qual),fac20));
;  312          0547  1
;  313        P 0548  1 $state(facil_qual,
;  314        P 0549  1           ('SHARED',tpa$_exit,,shared_mask,facility_flags),
;  315        P 0550  1           ('SYSTEM',tpa$_exit,,system_mask,facility_flags),
;  316        P 0551  1           ('PREFIX',fac_prefix),
;  317          0552  1           ('MACRO',fac_macro));
;  318          0553  1
;  319        P 0554  1 $state(fac_prefix,
;  320        P 0555  1           ('='),
;  321          0556  1           (':'));
;  322          0557  1
;  323        P 0558  1 $state(,
;  324        P 0559  1           (tpa$_symbol,tpa$_e it,store_string,,,
;  325          0560  1                   PLIT(default_prefix,1,defpre_bufsiz)));
;  326          0561  1
;  327        P 0562  1 $state(fac_macro,
;  328        P 0563  1           ('='),
;  329          0564  1           (':'));
;  330          0565  1
;  331        P 0566  1 $state(,
;  332        P 0567  1           (tpa$_symbol,tpa$_exit,store_string,,,
;  333          0568  1                   PLIT(macro_name,1,macro_bufsiz)));
;  334          0569  1
;  335          0570  1 !
;  336          0571  1 !       Parse .IDENT specification
;  337          0572  1 !
;  338          0573  1
;  339        P 0574  1 $state( ident1,
;  340        P 0575  1           ((contin),ident1,get_cont_line),
;  341        P 0576  1           (tpa$_symbol,ident2,build_version),
;  342        P 0577  1           ('"', ident2,find_endvers),
;  343        P 0578  1           ('''',ident2,find_endvers),
;  344          0579  1           ('/', ident2,find_endvers) );
;  345          0580  1
;  346          0581  1
;  347        P 0582  1 $state( ident2,
;  348        P 0583  1           ((contin),ident2,get_cont_line),
;  349          0584  1           (tpa$_lambda,end_line) );
;  350          0585  1
;  351          0586  1
;  352          0587  1 !
;  353          0588  1 !       Parse .SEVERITY command to set default severity
;  354          0589  1 !
;  355          0590  1
;  356        P 0591  1 $state(severity,
;  357        P 0592  1           ((contin),severity,get_cont_line),
;  358        P 0593  1           ((parse_severity),end_line,store_number,,,
;  359          0594  1                   PLIT(default_sev)));
;  360          0595  1
;  361          0596  1 !
;  362        0597  1 !       Parse .BASE command to set new message number
;  363        0598  1 !
;  364        0599  1
;  365      P 0600  1 $state(base,
;  366      P 0601  1         ((contin),base,get_cont_line),
;  367      P 0602  1         ((expression),end_line,store_number,,,
;  368        0603  1                 PLIT(message_number,0,1^$FIELDWIDTH(sts$v_code)-1)));
;  369        0604  1
;  370        0605  1
;  371        0606  1 !
;  372        0607  1 !       Parse .LITERAL command to define literals
;  373        0608  1 !
;  374        0609  1
;  375      P 0610  1 $state(literal_stmt,
;  376      P 0611  1         ((contin),literal_stmt,get_cont_line),
;  377        0612  1         (tpa$_lambda,,set_number,,,1));         ! Default 1st literal = 1
;  378        0613  1
;  379      P 0614  1 $state(,
;  380        0615  1         (tpa$_lambda,,store_number,,,PLIT(literal_value)));
;  381        0616  1
;  382      P 0617  1 $state(next_literal,
;  383      P 0618  1         ((contin),next_literal,get_cont_line),
;  384        0619  1         (tpa$_symbol,,,,literal_name));
;  385        0620  1
;  386      P 0621  1 $state(,
;  387      P 0622  1         ('='),
;  388      P 0623  1         (':'),
;  389        0624  1         (tpa$_lambda,set_literal));
;  390        0625  1
;  391      P 0626  1 $state(,
;  392        0627  1         ((expression),,store_number,,,PLIT(literal_value)));
;  393        0628  1
;  394      P 0629  1 $state(set_literal,
;  395        0630  1         (tpa$_lambda,,define_literal));
;  396        0631  1
;  397      P 0632  1 $state(end_literal,
;  398      P 0633  1         (',',next_literal),
;  399        0634  1         (tpa$_lambda,end_line));
;  400        0635  1
;  401        0636  1 !
;  402        0637  1 !       Parse .LANGUAGE command to set default language
;  403        0638  1 !
;  404        0639  1
;  405      P 0640  1 $state(language,
;  406      P 0641  1         ((contin),language,get_cont_line),
;  407      P 0642  1         ((parse_lang),end_line,store_number,,,
;  408        0643  1                 PLIT(default_lang)));
;  409        0644  1
;  410        0645  1 !
;  411        0646  1 !       Parse .TITLE command
;  412        0647  1 !
;  413        0648  1
;  414      P 0649  1 $state(title,
;  415      P 0650  1         ((contin),title,get_cont_line),
;  416        0651  1         (tpa$_symbol,,set_module));
;  417        0652  1
;  418      P 0653  1 $state(title2,
;  419        P 0654  1          ((contin),title2,get_cont_line),
;  420        P 0655  1          (tpa$_any,end_line,set_title),
;  421        P 0656  1 !        (tpa$_symbol,end_line,set_title),
;  422          0657  1          (tpa$_lambda,end_line));
;  423          0658  1
;  424          0659  1 !
;  425          0660  1 !        Parse message definition line
;  426          0661  1 !
;  427          0662  1
;  428        P 0663  1 $state(definition,
;  429        P 0664  1          ((contin),definition,get_cont_line),
;  430        P 0665  1          ('/'),
;  431        P 0666  1          ('<',def1,find_bracket),
;  432          0667  1          ('"',def1,find_eos));
;  433          0668  1
;  434        P 0669  1 $state(,
;  435          0670  1          ((def_qual),definition));
;  436          0671  1
;  437        P 0672  1 $state(def1,
;  438        P 0673  1          ((contin),def1,get_cont_line),
;  439        P 0674  1          ('/'),
;  440          0675  1          (tpa$_lambda,end_line,message_defn));
;  441          0676  1
;  442        P 0677  1 $state(,
;  443          0678  1          ((def_qual),def1));
;  444          0679  1
;  445        P 0680  1 $state(def_qual,
;  446        P 0681  1          ((parse_severity),tpa$_exit,store_number,,,
;  447        P 0682  1                  PLIT(severity_value)),
;  448        P 0683  1          ('FAO_COUNT',faocnt),
;  449        P 0684  1          ('IDENTIFICATION',ident),
;  450        P 0685  1          ('DETAIL',detail),
;  451        P 0686  1          ('LANGUAGE',lang),
;  452          0687  1          ('USER_VALUE',userval));
;  453          0688  1
;  454        P 0689  1 $state(faocnt,
;  455        P 0690  1          ('='),
;  456          0691  1          (':'));
;  457          0692  1
;  458        P 0693  1 $state(,
;  459        P 0694  1          ((expression),tpa$_exit,store_number,,,
;  460          0695  1                  PLIT(faocnt_value,0,31)));
;  461          0696  1
;  462        P 0697  1 $state(ident,
;  463        P 0698  1          ('='),
;  464          0699  1          (':'));
;  465          0700  1
;  466        P 0701  1 $state(,
;  467        P 0702  1          (tpa$_symbol,tpa$_exit,store_string,,,
;  468          0703  1                  PLIT(ident_value,1,ident_bufsiz)));
;  469          0704  1
;  470        P 0705  1 $state(detail,
;  471        P 0706  1          ('='),
;  472          0707  1          (':'));
;  473          0708  1
;  474        P 0709  1 $state(,
;  475        P 0710  1          ((expression),tpa$_exit,store_number,,,
    1                   PLIT(detail_value,0,255)));
    1
P   1 $state(lang,
P   1         ('='),
    1         (':'));
    1
P   1 $state(,
P   1         ((parse_lang),tpa$_exit,store_number,,,
    1                 PLIT(lang_value)));
    1
P   1 $state(userval,
P   1         ('='),
    1         (':'));
    1
P   1 $state(,
P   1         ((expression),tpa$_exit,store_number,,,
    1                 PLIT(userval_value,0,255)));
    1
    1 !
    1 !       Translate the language keyword into a number
    1 !
    1
P   1 $state(parse_lang,
P   1         ('ENGLISH',tpa$_exit,set_number,,,mrec$c_english),
P   1         ('FRENCH',tpa$_exit,set_number,,,mrec$c_french),
    1         ('GERMAN',tpa$_exit,set_number,,,mrec$c_german));
    1
    1 !
    1 !       Translate the SEVERITY keyword into the severity number
    1 !
    1
P   1 $state(parse_severity,
P   1         ('FATAL',tpa$_exit,set_number,,,sts$k_severe),
P   1         ('SEVERE',tpa$_exit,set_number,,,sts$k_severe),
P   1         ('INFORMATIONAL',tpa$_exit,set_number,,,sts$k_info),
P   1         ('SUCCESS',tpa$_exit,set_number,,,sts$k_success),
P   1         ('ERROR',tpa$_exit,set_number,,,sts$k_error),
    1         ('WARNING',tpa$_exit,set_number,,,sts$k_warning));
    1
    1 ROUTINE null2: NOVALUE =;


                                                        .PSECT  _LIB$KEY1$,NOWRT,  SHR,  PIC,1

                                          00000 ;TPA$KEYST0
                                                U.25:   .BLKB   0
        59  54  49  4C  49  43  41  46   00000 ;TPA$KEYST
                                                U.27:   .ASCII  \FACILITY\                         ;
                                     FF  00008          .BYTE   -1                                 ;
                                          00009 ;TPA$KEYST0
                                                U.32:   .BLKB   0
        59  54  49  52  45  56  45  53   00009 ;TPA$KEYST
                                                U.34:   .ASCII  \SEVERITY\                         ;
                                     FF  00011          .BYTE   -1                                 ;
                                          00012 ;TPA$KEYST0
                                                U.38:   .BLKB   0
        45 47 41 55 47 4E 41 4C  00012 ;TPA$KEYST
                                        U.40:    .ASCII  \LANGUAGE\                              ;
                              FF 0001A           .BYTE   -1                                      ;
                                 0001B ;TPA$KEYSTO
                                        U.44:    .BLKB   0
                    54 4E 45 44 49  0001B ;TPA$KEYST
                                        U.46:    .ASCII  \IDENT\                                 ;
                              FF 00020           .BYTE   -1                                      ;
                                 00021 ;TPA$KEYSTO
                                        U.50:    .BLKB   0
                       45 53 41 42  00021 ;TPA$KEYST
                                        U.52:    .ASCII  \BASE\                                  ;
                              FF 00025           .BYTE   -1                                      ;
                                 00026 ;TPA$KEYSTO
                                        U.56:    .BLKB   0
           4C 41 52 45 54 49 4C  00026 ;TPA$KEYST
                                        U.58:    .ASCII  \LITERAL\                               ;
                              FF 0002D           .BYTE   -1                                      ;
                                 0002E ;TPA$KEYSTO
                                        U.62:    .BLKB   0
                       45 47 41 50  0002E ;TPA$KEYST
                                        U.64:    .ASCII  \PAGE\                                  ;
                              FF 00032           .BYTE   -1                                      ;
                                 00033 ;TPA$KEYSTO
                                        U.69:    .BLKB   0
                    45 4C 54 49 54  00033 ;TPA$KEYST
                                        U.71:    .ASCII  \TITLE\                                 ;
                              FF 00038           .BYTE   -1                                      ;
                                 00039 ;TPA$KEYSTO
                                        U.75:    .BLKB   0
                          44 4E 45  00039 ;TPA$KEYST
                                        U.77:    .ASCII  \END\                                   ;
                              FF 0003C           .BYTE   -1                                      ;
                              FF 0003D ;TPA$KEYFILL
                                        U.81:    .BYTE   -1                                      ;
                                 0003E ;TPA$KEYSTO
                                        U.140:   .BLKB   0
              44 45 52 41 48 53  0003E ;TPA$KEYST
                                        U.142:   .ASCII  \SHARED\                                ;
                              FF 00044           .BYTE   -1                                      ;
                                 00045 ;TPA$KEYSTO
                                        U.147:   .BLKB   0
              4D 45 54 53 59 53  00045 ;TPA$KEYST
                                        U.149:   .ASCII  \SYSTEM\                                ;
                              FF 0004B           .BYTE   -1                                      ;
                                 0004C ;TPA$KEYSTO
                                        U.154:   .BLKB   0
              58 49 46 45 52 50  0004C ;TPA$KEYST
                                        U.156:   .ASCII  \PREFIX\                                ;
                              FF 00052           .BYTE   -1                                      ;
                                 00053 ;TPA$KEYSTO
                                        U.160:   .BLKB   0
                    4F 52 43 41 4D  00053 ;TPA$KEYST
                                        U.162:   .ASCII  \MACRO\                                 ;
                              FF 00058           .BYTE   -1                                      ;
                              FF 00059 ;TPA$KEYFILL
                                        U.166:   .BYTE   -1                                      ;
```

```
                                                   0005A ;TPA$KEYST0
                                                         U.318:  .BLKB   0
                     54  4E  55  4F  43  5F  4F  41  46  0005A ;TPA$KEYST
                                                         U.320:  .ASCII  \FAO_COUNT\                              ;
                                                     FF  00063           .BYTE   -1                               ;
                                                         00064 ;TPA$KEYST0
                                                         U.324:  .BLKB   0
     4E  4F  49  54  41  43  49  46  49  54  4E  45  44  49  00064 ;TPA$KEYST
                                                         U.326:  .ASCII  \IDENTIFICATION\                         ;
                                                     FF  00072           .BYTE   -1                               ;
                                                         00073 ;TPA$KEYST0
                                                         U.330:  .BLKB   0
                                 4C  49  41  54  45  44  00073 ;TPA$KEYST
                                                         U.332:  .ASCII  \DETAIL\                                 ;
                                                     FF  00079           .BYTE   -1                               ;
                                                         0007A ;TPA$KEYST0
                                                         U.336:  .BLKB   0
                         45  47  41  55  47  4E  41  4C  0007A ;TPA$KEYST
                                                         U.338:  .ASCII  \LANGUAGE\                               ;
                                                     FF  00082           .BYTE   -1                               ;
                                                         00083 ;TPA$KEYST0
                                                         U.342:  .BLKB   0
                 45  55  4C  41  56  5F  52  45  53  55  00083 ;TPA$KEYST
                                                         U.344:  .ASCII  \USER_VALUE\                             ;
                                                     FF  0008D           .BYTE   -1                               ;
                                                     FF  0008E ;TPA$KEYFILL
                                                         U.348:  .BYTE   -1                               ;
                                                         0008F ;TPA$KEYST0
                                                         U.388:  .BLKB   0
                             48  53  49  4C  47  4E  45  0008F ;TPA$KEYST
                                                         U.390:  .ASCII  \ENGLISH\                                ;
                                                     FF  00096           .BYTE   -1                               ;
                                                         00097 ;TPA$KEYST0
                                                         U.396:  .BLKB   0
                                 48  43  4E  45  52  46  00097 ;TPA$KEYST
                                                         U.398:  .ASCII  \FRENCH\                                 ;
                                                     FF  0009D           .BYTE   -1                               ;
                                                         0009E ;TPA$KEYST0
                                                         U.404:  .BLKB   0
                                 4E  41  4D  52  45  47  0009E ;TPA$KEYST
                                                         U.406:  .ASCII  \GERMAN\                                 ;
                                                     FF  000A4           .BYTE   -1                               ;
                                                     FF  000A5 ;TPA$KEYFILL
                                                         U.412:  .BYTE   -1                               ;
                                                         000A6 ;TPA$KEYST0
                                                         U.413:  .BLKB   0
                                     4C  41  54  41  46  000A6 ;TPA$KEYST
                                                         U.415:  .ASCII  \FATAL\                                  ;
                                                     FF  000AB           .BYTE   -1                               ;
                                                         000AC ;TPA$KEYST0
                                                         U.421:  .BLKB   0
                                 45  52  45  56  45  53  000AC ;TPA$KEYST
                                                         U.423:  .ASCII  \SEVERE\                                 ;
                                                     FF  000B2           .BYTE   -1                               ;
                                                         000B3 ;TPA$KEYST0
                                                         U.429:  .BLKB   0
         4C  41  4E  4F  49  54  41  4D  52  4F  46  4E  49  000B3 ;TPA$KEYST
```

```
                                   U.431:  .ASCII  \INFORMATIONAL\                                ;
                       FF  000C0           .BYTE   -1                                             ;
                           000C1 ;TPA$KEYST0
                                   U.437:  .BLKB   0
  53  53  45  43  43  55  53  000C1 ;TPA$KEYST
                                   U.439:  .ASCII  \SUCCESS\                                      ;
                       FF  000C8           .BYTE   -1                                             ;
                           000C9 ;TPA$KEYST0
                                   U.445:  .BLKB   0
          52  4F  52  52  45  000C9 ;TPA$KEYST
                                   U.447:  .ASCII  \ERROR\                                        ;
                       FF  000CE           .BYTE   -1                                             ;
                           000CF ;TPA$KEYST0
                                   U.453:  .BLKB   0
  47  4E  49  4E  52  41  57  000CF ;TPA$KEYST
                                   U.455:  .ASCII  \WARNING\                                      ;
                       FF  000D6           .BYTE   -1                                             ;
                       FF  000D7 ;TPA$KEYFILL
                                   U.461:  .BYTE   -1                                             ;

                                           .PSECT  _LIB$STATE$,NOWRT,  SHR,  PIC,1

                           00000 PARSE_STATES::
                                           .BLKB   0
                     85F6  00000 ;TPA$TYPE
                                   U.2:    .WORD   -31242                                         ;
                00000000V  00002 ;TPA$ACTION
                                   U.3:    .LONG   <<INIT_STACK-U.3>-4>                           ;
                     85F6  00006 ;TPA$TYPE
                                   U.4:    .WORD   -31242                                         ;
                00000000V  00008 ;TPA$ACTION
                                   U.5:    .LONG   <<MESSAGE_INIT-U.5>-4>                         ;
                           0000C MAIN:     .BLKB   0
                     99F8  0000C ;TPA$TYPE
                                   U.6:    .WORD   -26120                                         ;
                    0000*  0000E ;TPA$SUBEXP
                                   U.8:    .WORD   <<U.7-U.8>-2>                                  ;
                00000000V  00010 ;TPA$ACTION
                                   U.9:    .LONG   <<GET_CONT_LINE-U.9>-4>                        ;
                    0000*  00014 ;TPA$TARGET
                                   U.10.   .WORD   <<MAIN-U.10>-2>                                ;
                     9021  00016 ;TPA$TYPE
                                   U.11:   .WORD   -28639                                         ;
                00000000V  00018 ;TPA$ACTION
                                   U.12:   .LONG   <<COMMENT-U.12>-4>                             ;
                     FFFF  0001C ;TPA$TARGET
                                   U.13:   .WORD   -1                                             ;
                     11F7  0001E ;TPA$TYPE
                                   U.14:   .WORD   4599                                           ;
                     FFFF  00020 ;TPA$TARGET
                                   U.15:   .WORD   -1                                             ;
                     102E  00022 ;TPA$TYPE
                                   U.16:   .WORD   4142                                           ;
                    0000*  00024 ;TPA$TARGET
                                   U.18:   .WORD   <<U.17-U.18>-2>                                ;
                     97F1  00026 ;TPA$TYPE
                                   U.19:   .WORD   -26639                                         ;
```

```
               01  00028 ;TPA$FLAGS2
                         U.20:     .BYTE    1                                         ;
       00000000'  00029 ;TPA$PARAM
                         U.21:     .ADDRESS P.AAA                                     ;
       00000000V  0002D ;TPA$ACTION
                         U.22:     .LONG    <<STORE_STRING-U.22>-4>                   ;
            0000*  00031 ;TPA$TARGET
                         U.24:     .WORD    <<U.23-U.24>-2>                           ;
                   00033 ;DIRECTIVE
                         U.17:     .BLKB    0
            9100   00033 ;TPA$TYPE
                         U.28:     .WORD    -28416                                    ;
       00000000V  00035 ;TPA$ACTION
                         U.29:     .LONG    <<FACILITY_INIT-U.29>-4>                  ;
            0000*  00039 ;TPA$TARGET
                         U.31:     .WORD    <<U.30-U.31>-2>                           ;
            1101   0003B ;TPA$TYPE
                         U.35:     .WORD    4353                                      ;
            0000*  0003D ;TPA$TARGET
                         U.37:     .WORD    <<U.36-U.37>-2>                           ;
            1102   0003F ;TPA$TYPE
                         U.41:     .WORD    4354                                      ;
            0000*  00041 ;TPA$TARGET
                         U.43:     .WORD    <<U.42-U.43>-2>                           ;
            1103   00043 ;TPA$TYPE
                         U.47:     .WORD    4355                                      ;
            0000*  00045 ;TPA$TARGET
                         U.49:     .WORD    <<U.48-U.49>-2>                           ;
            1104   00047 ;TPA$TYPE
                         U.53:     .WORD    4356                                      ;
            0000*  00049 ;TPA$TARGET
                         U.55:     .WORD    <<U.54-U.55>-2>                           ;
            1105   0004B ;TPA$TYPE
                         U.59:     .WORD    4357                                      ;
            0000*  0004D ;TPA$TARGET
                         U.61:     .WORD    <<U.60-U.61>-2>                           ;
            9106   0004F ;TPA$TYPE
                         U.65:     .WORD    -28410                                    ;
       00000000*  00051 ;TPA$ACTION
                         U.66:     .LONG    <<NEW_PAGE-U.66>-4>                       ;
            0000*  00055 ;TPA$TARGET
                         U.68:     .WORD    <<U.67-U.68>-2>                           ;
            1107   00057 ;TPA$TYPE
                         U.72:     .WORD    4359                                      ;
            0000*  00059 ;TPA$TARGET
                         U.74:     .WORD    <<U.73-U.74>-2>                           ;
            9508   0005B ;TPA$TYPE
                         U.78:     .WORD    -27384                                    ;
       00000000V  0005D ;TPA$ACTION
                         U.79:     .LONG    <<FACILITY_INIT-U.79>-4>                  ;
            0000*  00061 ;TPA$TARGET
                         U.80:     .WORD    <<U.67-U.80>-2>                           ;
                   00063 ;CONTIN
                         U.7:      .BLKB    0
            042D   00063 ;TPA$TYPE
                         U.82:     .WORD    1069                                      ;
            9021   00065 ;TPA$TYPE
```

```
                          U.83:     .WORD    -28639                                  ;
00000000V 00067 ;TPA$ACTION
                          U.84:     .LONG    <<COMMENT-U.84>-4>                      ;
     FFFF 0006B ;TPA$TARGET
                          U.85:     .WORD    -1                                      ;
     15F7 0006D ;TPA$TYPE
                          U.86:     .WORD    5623                                    ;
     FFFF 0006F ;TPA$TARGET
                          U.87:     .WORD    -1                                      ;
          00071 ;END_LINE
                          U.67:     .BLKB    0
     99F8 00071 ;TPA$TYPE
                          U.88:     .WORD    -26120                                  ;
    0000* 00073 ;TPA$SUBEXP
                          U.89:     .WORD    <<U.7-U.89>-2>                          ;
00000000V 00075 ;TPA$ACTION
                          U.90:     .LONG    <<GET_CONT_LINE-U.90>-4>                ;
    0000* 00079 ;TPA$TARGET
                          U.91:     .WORD    <<U.67-U.91>-2>                         ;
     9021 0007B ;TPA$TYPE
                          U.92:     .WORD    -28639                                  ;
00000000V 0007D ;TPA$ACTION
                          U.93:     .LONG    <<COMMENT-U.93>-4>                      ;
     FFFF 00081 ;TPA$TARGET
                          U.94:     .WORD    -1                                      ;
     15F7 00083 ;TPA$TYPE
                          U.95:     .WORD    5623                                    ;
     FFFF 00085 ;TPA$TARGET
                          U.96:     .WORD    -1                                      ;
          00087 ;FACILITY
                          U.30:     .BLKB    0
     99F8 00087 ;TPA$TYPE
                          U.97:     .WORD    -26120                                  ;
    0000* 00089 ;TPA$SUBEXP
                          U.98:     .WORD    <<U.7-U.98>-2>                          ;
00000000V 0008B ;TPA$ACTION
                          U.99:     .LONG    <<GET_CONT_LINE-U.99>-4>                ;
    0000* 0008F ;TPA$TARGET
                          U.100:    .WORD    <<U.30-U.100>-2>                        ;
     002F 00091 ;TPA$TYPE
                          U.101:    .WORD    47                                      ;
     15F6 00093 ;TPA$TYPE
                          U.102:    .WORD    5622                                    ;
    0000* 00095 ;TPA$TARGET
                          U.104:    .WORD    <<U.103-U.104>-2>                       ;
     1DF8 00097 ;TPA$TYPE
                          U.105:    .WORD    7672                                    ;
    0000* 00099 ;TPA$SUBEXP
                          U.107:    .WORD    <<U.106-U.107>-2>                       ;
    0000* 0009B ;TPA$TARGET
                          U.108:    .WORD    <<U.30-U.108>-2>                        ;
          0009D ;FAC10
                          U.103:    .BLKB    0
     87F1 0009D ;TPA$TYPE
                          U.109:    .WORD    -30735                                  ;
       01 0009F ;TPA$FLAGS2
                          U.110:    .BYTE    1                                       ;
```

```
 00000000'  000A0  ;TPA$PARAM
                   U.111:   .ADDRESS P.AAB                                    ;
 00000000V  000A4  ;TPA$ACTION
                   U.112:   .LONG    <<STORE_STRING-U.112>-4>                 ;
            000A8  FAC15:   .BLKB    0
      99F8  000A8  ;TPA$TYPE
                   U.113:   .WORD    -26120                                   ;
     0000*  000AA  ;TPA$SUBEXP
                   U.114:   .WORD    <<U.7-U.114>-2>                          ;
 00000000V  000AC  ;TPA$ACTION
                   U.115:   .LONG    <<GET_CONT_LINE-U.115>-4>                ;
     0000*  000B0  ;TPA$TARGET
                   U.116:   .WORD    <<FAC15-U.116>-2>                        ;
      002C  000B2  ;TPA$TYPE
                   U.117:   .WORD    44                                       ;
      05F6  000B4  ;TPA$TYPE
                   U.118:   .WORD    1526                                     ;
            000B6  FAC18:   .BLKB    0
      99F8  000B6  ;TPA$TYPE
                   U.119:   .WORD    -26120                                   ;
     0000*  000B8  ;TPA$SUBEXP
                   U.120:   .WORD    <<U.7-U.120>-2>                          ;
 00000000V  000BA  ;TPA$ACTION
                   U.121:   .LONG    <<GET_CONT_LINE-U.121>-4>                ;
     0000*  000BE  ;TPA$TARGET
                   U.122:   .WORD    <<FAC18-U.122>-2>                        ;
      8FF8  000C0  ;TPA$TYPE
                   U.123:   .WORD    -28680                                   ;
        01  000C2  ;TPA$FLAGS2
                   U.124:   .BYTE    1                                        ;
     0000V  000C3  ;TPA$SUBEXP
                   U.126:   .WORD    <<U.125-U.126>-2>                        ;
 00000000'  000C5  ;TPA$PARAM
                   U.127:   .ADDRESS P.AAC                                    ;
 00000000V  000C9  ;TPA$ACTION
                   U.128:   .LONG    <<STORE_NUMBER-U.128>-4>                 ;
            000CD  FAC20:   .BLKB    0
      99F8  000CD  ;TPA$TYPE
                   U.129:   .WORD    -26120                                   ;
     0000*  000CF  ;TPA$SUBEXP
                   U.130:   .WORD    <<U.7-U.130>-2>                          ;
 00000000V  000D1  ;TPA$ACTION
                   U.131:   .LONG    <<GET_CONT_LINE-U.131>-4>                ;
     0000*  000D5  ;TPA$TARGET
                   U.132:   .WORD    <<FAC20-U.132>-2>                        ;
      002F  000D7  ;TPA$TYPE
                   U.133:   .WORD    47                                       ;
      95F6  000D9  ;TPA$TYPE
                   U.134:   .WORD    -27146                                   ;
 00000000V  000DB  ;TPA$ACTION
                   U.135:   .LONG    <<FACILITY_DEFN-U.135>-4>                ;
     0000*  000DF  ;TPA$TARGET
                   U.136:   .WORD    <<U.67-U.136>-2>                         ;
      1DF8  000E1  ;TPA$TYPE
                   U.137:   .WORD    7672                                     ;
     0000*  000E3  ;TPA$SUBEXP
                   U.138:   .WORD    <<U.106-U.138>-2>                        ;
     0000* 000E5 ;TPA$TARGET
                 U.139:  .WORD   <<FAC20-U.139>-2>                                  ;
           000E7 ;FACIL_QUAL
                 U.106:  .BLKB   0
     7109  000E7 ;TPA$TYPE
                 U.143:  .WORD   28937                                              ;
 00000000* 000E9 ;TPA$ADDR
                 U.144:  .LONG   <<FACILITY_FLAGS-U.144>-4>                         ;
 00000001  000ED ;TPA$MASK
                 U.145:  .LONG   1                                                  ;
     FFFF  000F1 ;TPA$TARGET
                 U.146:  .WORD   -1                                                 ;
     710A  000F3 ;TPA$TYPE
                 U.150:  .WORD   28938                                              ;
 00000000* 000F5 ;TPA$ADDR
                 U.151:  .LONG   <<FACILITY_FLAGS-U.151>-4>                         ;
 00000002  000F9 ;TPA$MASK
                 U.152:  .LONG   2                                                  ;
     FFFF  000FD ;TPA$TARGET
                 U.153:  .WORD   -1                                                 ;
     110B  000FF ;TPA$TYPE
                 U.157:  .WORD   4363                                               ;
     0000* 00101 ;TPA$TARGET
                 U.159:  .WORD   <<U.158-U.159>-2>                                  ;
     150C  00103 ;TPA$TYPE
                 U.163:  .WORD   5388                                               ;
     0000* 00105 ;TPA$TARGET
                 U.165:  .WORD   <<U.164-U.165>-2>                                  ;
           00107 ;FAC_PREFIX
                 U.158:  .BLKB   0
     003D  00107 ;TPA$TYPE
                 U.167:  .WORD   61                                                 ;
     043A  00109 ;TPA$TYPE
                 U.168:  .WORD   1082                                               ;
     97F1  0010B ;TPA$TYPE
                 U.169:  .WORD   -26639                                             ;
       01  0010D ;TPA$FLAGS2
                 U.170:  .BYTE   1                                                  ;
 00000000' 0010E ;TPA$PARAM
                 U.171:  .ADDRESS P.AAD                                             ;
 00000000V 00112 ;TPA$ACTION
                 U.172:  .LONG   <<STORE_STRING-U.172>-4>                           ;
     FFFF  00116 ;TPA$TARGET
                 U.173:  .WORD   -1                                                 ;
           00118 ;FAC_MACRO
                 U.164:  .BLKB   0
     003D  00118 ;TPA$TYPE
                 U.174:  .WORD   61                                                 ;
     043A  0011A ;TPA$TYPE
                 U.175:  .WORD   1082                                               ;
     97F1  0011C ;TPA$TYPE
                 U.176:  .WORD   -26639                                             ;
       01  0011E ;TPA$FLAGS2
                 U.177:  .BYTE   1                                                  ;
 00000000' 0011F ;TPA$PARAM
                 U.178:  .ADDRESS P.AAE                                             ;
 00000000V 00123 ;TPA$ACTION
```

```
                           U.179:  .LONG    <<STORE_STRING-U.179>-4>              ;
        FFFF  00127 ;TPA$TARGET
                           U.180:  .WORD    -1                                    ;
              00129 ;IDENT1
                           U.48:   .BLKB    0
        99F8  00129 ;TPA$TYPE
                           U.181:  .WORD    -26120                                ;
       0000*  0012B ;TPA$SUBEXP
                           U.182:  .WORD    <<U.7-U.182>-2>                       ;
   00000000V  0012D ;TPA$ACTION
                           U.183:  .LONG    <<GET_CONT_LINE-U.183>-4>             ;
       0000*  00131 ;TPA$TARGET
                           U.184:  .WORD    <<U.48-U.184>-2>                      ;
        91F1  00133 ;TPA$TYPE
                           U.185:  .WORD    -28175                                ;
   00000000V  00135 ;TPA$ACTION
                           U.186:  .LONG    <<BUILD_VERSION-U.186>-4>             ;
       0000*  00139 ;TPA$TARGET
                           U.188:  .WORD    <<U.187-U.188>-2>                     ;
        9022  0013B ;TPA$TYPE
                           U.189:  .WORD    -28638                                ;
   00000000V  0013D ;TPA$ACTION
                           U.190:  .LONG    <<FIND_ENDVERS-U.190>-4>              ;
       0000*  00141 ;TPA$TARGET
                           U.191:  .WORD    <<U.187-U.191>-2>                     ;
        9027  00143 ;TPA$TYPE
                           U.192:  .WORD    -28633                                ;
   00000000V  00145 ;TPA$ACTION
                           U.193:  .LONG    <<FIND_ENDVERS-U.193>-4>              ;
       0000*  00149 ;TPA$TARGET
                           U.194:  .WORD    <<U.187-U.194>-2>                     ;
        942F  0014B ;TPA$TYPE
                           U.195:  .WORD    -27601                                ;
   00000000V  0014D ;TPA$ACTION
                           U.196:  .LONG    <<FIND_ENDVERS-U.196>-4>              ;
       0000*  00151 ;TPA$TARGET
                           U.197:  .WORD    <<U.187-U.197>-2>                     ;
              00153 ;IDENT2
                           U.187:  .BLKB    0
        99F8  00153 ;TPA$TYPE
                           U.198:  .WORD    -26120                                ;
       0000*  00155 ;TPA$SUBEXP
                           U.199:  .WORD    <<U.7-U.199>-2>                       ;
   00000000V  00157 ;TPA$ACTION
                           U.200:  .LONG    <<GET_CONT_LINE-U.200>-4>             ;
       0000*  0015B ;TPA$TARGET
                           U.201:  .WORD    <<U.187-U.201>-2>                     ;
        15F6  0015D ;TPA$TYPE
                           U.202:  .WORD    5622                                  ;
       0000*  0015F ;TPA$TARGET
                           U.203:  .WORD    <<U.67-U.203>-2>                      ;
              00161 ;SEVERITY
                           U.36:   .BLKB    0
        99F8  00161 ;TPA$TYPE
                           U.204:  .WORD    -26120                                ;
       0000*  00163 ;TPA$SUBEXP
                           U.205:  .WORD    <<U.7-U.205>-2>                       ;
```

```
 00000000V 00165 ;TPA$ACTION
                 U.206:  .LONG   <<GET_CONT_LINE-U.206>-4>                  ;
     0000* 00169 ;TPA$TARGET
                 U.207:  .WORD   <<U.36-U.207>-2>                           ;
     9FF8  0016B ;TPA$TYPE
                 U.208:  .WORD   -24584                                     ;
       01  0016D ;TPA$FLAGS2
                 U.209:  .BYTE   1                                          ;
     0000* 0016E ;TPA$SUBEXP
                 U.211:  .WORD   <<U.210-U.211>-2>                          ;
 00000000' 00170 ;TPA$PARAM
                 U.212:  .ADDRESS P.AAF                                     ;
 00000000V 00174 ;TPA$ACTION
                 U.213:  .LONG   <<STORE_NUMBER-U.213>-4>                   ;
     0000* 00178 ;TPA$TARGET
                 U.214:  .WORD   <<U.67-U.214>-2>                           ;
           0017A ;BASE
                 U.54:   .BLKB   0
     99F8  0017A ;TPA$TYPE
                 U.215:  .WORD   -26120                                     ;
     0000* 0017C ;TPA$SUBEXP
                 U.216:  .WORD   <<U.7-U.216>-2>                            ;
 00000000V 0017E ;TPA$ACTION
                 U.217:  .LONG   <<GET_CONT_LINE-U.217>-4>                  ;
     0000* 00182 ;TPA$TARGET
                 U.218:  .WORD   <<U.54-U.218>-2>                           ;
     9FF8  00184 ;TPA$TYPE
                 U.219:  .WORD   -24584                                     ;
       01  00186 ;TPA$FLAGS2
                 U.220:  .BYTE   1                                          ;
     0000V 00187 ;TPA$SUBEXP
                 U.221:  .WORD   <<U.125-U.221>-2>                          ;
 00000000' 00189 ;TPA$PARAM
                 U.222:  .ADDRESS P.AAG                                     ;
 00000000V 0018D ;TPA$ACTION
                 U.223:  .LONG   <<STORE_NUMBER-U.223>-4>                   ;
     0000* 00191 ;TPA$TARGET
                 U.224:  .WORD   <<U.67-U.224>-2>                           ;
           00193 ;LITERAL_STMT
                 U.60:   .BLKB   0
     99F8  00193 ;TPA$TYPE
                 U.225:  .WORD   -26120                                     ;
     0000* 00195 ;TPA$SUBEXP
                 U.226:  .WORD   <<U.7-U.226>-2>                            ;
 00000000V 00197 ;TPA$ACTION
                 U.227:  .LONG   <<GET_CONT_LINE-U.227>-4>                  ;
     0000* 0019B ;TPA$TARGET
                 U.228:  .WORD   <<U.60-U.228>-2>                           ;
     87F6  0019D ;TPA$TYPE
                 U.229:  .WORD   -30730                                     ;
       01  0019F ;TPA$FLAGS2
                 U.230:  .BYTE   1                                          ;
 00000001  001A0 ;TPA$PARAM
                 U.231:  .LONG   1                                          ;
 00000000* 001A4 ;TPA$ACTION
                 U.232:  .LONG   <<SET_NUMBER-U.232>-4>                     ;
     87F6  001A8 ;TPA$TYPE
```

```
                            U.233:  .WORD   -30730                              ;
         01  001AA ;TPA$FLAGS2
                            U.234:  .BYTE   1                                   ;
  00000000'  001AB ;TPA$PARAM
                            U.235:  .ADDRESS P.AAH                              ;
  00000000V  001AF ;TPA$ACTION
                            U.236:  .LONG   <<STORE_NUMBER-U.236>-4>            ;
             001B3 NEXT_LITERAL:
                                    .BLKB   0
       99F8  001B3 ;TPA$TYPE
                            U.237:  .WORD   -26120                              ;
      0000*  001B5 ;TPA$SUBEXP
                            U.238:  .WORD   <<U.7-U.238>-2>                     ;
  00000000V  001B7 ;TPA$ACTION
                            U.239:  .LONG   <<GET_CONT_LINE-U.239>-4>           ;
      0000*  001BB ;TPA$TARGET
                            U.240:  .WORD   <<NEXT_LITERAL-U.240>-2>            ;
       45F1  001BD ;TPA$TYPE
                            U.241:  .WORD   17905                               ;
  00000000*  001BF ;TPA$ADDR
                            U.242:  .LONG   <<LITERAL_NAME-U.242>-4>            ;
       003D  001C3 ;TPA$TYPE
                            U.243:  .WORD   61                                  ;
       003A  001C5 ;TPA$TYPE
                            U.244:  .WORD   58                                  ;
       15F6  001C7 ;TPA$TYPE
                            U.245:  .WORD   5622                                ;
      0000*  001C9 ;TPA$TARGET
                            U.247:  .WORD   <<U.246-U.247>-2>                   ;
       8FF8  001CB ;TPA$TYPE
                            U.248:  .WORD   -28680                              ;
         01  001CD ;TPA$FLAGS2
                            U.249:  .BYTE   1                                   ;
      0000V  001CE ;TPA$SUBEXP
                            U.250:  .WORD   <<U.125-U.250>-2>                   ;
  00000000'  001D0 ;TPA$PARAM
                            U.251:  .ADDRESS P.AAI                              ;
  00000000V  001D4 ;TPA$ACTION
                            U.252:  .LONG   <<STORE_NUMBER-U.252>-4>            ;
             001D8 ;SET_LITERAL
                            U.246:  .BLKB   0
       85F6  001D8 ;TPA$TYPE
                            U.253:  .WORD   -31242                              ;
  00000000V  001DA ;TPA$ACTION
                            U.254:  .LONG   <<DEFINE_LITERAL-U.254>-4>          ;
             001DE END_LITERAL:
                                    .BLKB   0
       102C  001DE ;TPA$TYPE
                            U.255:  .WORD   4140                                ;
      0000*  001E0 ;TPA$TARGET
                            U.256:  .WORD   <<NEXT_LITERAL-U.256>-2>            ;
       15F6  001E2 ;TPA$TYPE
                            U.257:  .WORD   5622                                ;
      0000*  001E4 ;TPA$TARGET
                            U.258:  .WORD   <<U.67-U.258>-2>                    ;
             001E6 ;LANGUAGE
                            U.42:   .BLKB   0
         99F8  001E6 ;TPA$TYPE
                     U.259:   .WORD     -26120                                  ;
        0000*  001E8 ;TPA$SUBEXP
                     U.260:   .WORD     <<U.7-U.260>-2>                         ;
    00000000V  001EA ;TPA$ACTION
                     U.261:   .LONG     <<GET_CONT_LINE-U.261>-4>               ;
        0000*  001EE ;TPA$TARGET
                     U.262:   .WORD     <<U.42-U.262>-2>                        ;
         9FF8  001F0 ;TPA$TYPE
                     U.263:   .WORD     -24584                                  ;
           01  001F2 ;TPA$FLAGS2
                     U.264:   .BYTE     1                                       ;
        0000*  001F3 ;TPA$SUBEXP
                     U.266:   .WORD     <<U.265-U.266>-2>                       ;
    00000000'  001F5 ;TPA$PARAM
                     U.267:   .ADDRESS P.AAJ                                    ;
    00000000V  001F9 ;TPA$ACTION
                     U.268:   .LONG     <<STORE_NUMBER-U.268>-4>                ;
        0000*  001FD ;TPA$TARGET
                     U.269:   .WORD     <<U.67-U.269>-2>                        ;
               001FF ;TITLE
                     U.73:    .BLKB     0
         99F8  001FF ;TPA$TYPE
                     U.270:   .WORD     -26120                                  ;
        0000*  00201 ;TPA$SUBEXP
                     U.271:   .WORD     <<U.7-U.271>-2>                         ;
    00000000V  00203 ;TPA$ACTION
                     U.272:   .LONG     <<GET_CONT_LINE-U.272>-4>               ;
        0000*  00207 ;TPA$TARGET
                     U.273:   .WORD     <<U.73-U.273>-2>                        ;
         85F1  00209 ;TPA$TYPE
                     U.274:   .WORD     -31247                                  ;
    00000000V  0020B ;TPA$ACTION
                     U.275:   .LONG     <<SET_MODULE-U.275>-4>                  ;
               0020F TITLE2:  .BLKB     0
         99F8  0020F ;TPA$TYPE
                     U.276:   .WORD     -26120                                  ;
        0000*  00211 ;TPA$SUBEXP
                     U.277:   .WORD     <<U.7-U.277>-2>                         ;
    00000000V  00213 ;TPA$ACTION
                     U.278:   .LONG     <<GET_CONT_LINE-U.278>-4>               ;
        0000*  00217 ;TPA$TARGET
                     U.279:   .WORD     <<TITLE2-U.279>-2>                      ;
         91ED  00219 ;TPA$TYPE
                     U.280:   .WORD     -28179                                  ;
    00000000V  0021B ;TPA$ACTION
                     U.281:   .LONG     <<SET_TITLE-U.281>-4>                   ;
        0000*  0021F ;TPA$TARGET
                     U.282:   .WORD     <<U.67-U.282>-2>                        ;
         15F6  00221 ;TPA$TYPE
                     U.283:   .WORD     5622                                    ;
        0000*  00223 ;TPA$TARGET
                     U.284:   .WORD     <<U.67-U.284>-2>                        ;
               00225 ;DEFINITION
                     U.23:    .BLKB     0
         99F8  00225 ;TPA$TYPE
                     U.285:   .WORD     -26120                                  ;
```

```
     0000* 00227 ;TPA$SUBEXP
                 U.286:   .WORD     <<U.7-U.286>-2>                          ;
 00000000V 00229 ;TPA$ACTION
                 U.287:   .LONG     <<GET_CONT_LINE-U.287>-4>                ;
     0000* 0022D ;TPA$TARGET
                 U.288:   .WORD     <<U.23-U.288>-2>                         ;
     002F  0022F ;TPA$TYPE
                 U.289:   .WORD     47                                       ;
     903C  00231 ;TPA$TYPE
                 U.290:   .WORD     -28612                                   ;
 00000000* 00233 ;TPA$ACTION
                 U.291:   .LONG     <<FIND_BRACKET-U.291>-4>                 ;
     0000* 00237 ;TPA$TARGET
                 U.293:   .WORD     <<U.292-U.293>-2>                        ;
     9422  00239 ;TPA$TYPE
                 U.294:   .WORD     -27614                                   ;
 00000000V 0023B ;TPA$ACTION
                 U.295:   .LONG     <<FIND_EOS-U.295>-4>                     ;
     0000* 0023F ;TPA$TARGET
                 U.296:   .WORD     <<U.292-U.296>-2>                        ;
     1DF8  00241 ;TPA$TYPE
                 U.297:   .WORD     7672                                     ;
     0000* 00243 ;TPA$SUBEXP
                 U.299:   .WORD     <<U.298-U.299>-2>                        ;
     0000* 00245 ;TPA$TARGET
                 U.300:   .WORD     <<U.23-U.300>-2>                         ;
           00247 ;DEF1
                 U.292:   .BLKB     0
     99F8  00247 ;TPA$TYPE
                 U.301:   .WORD     -26120                                   ;
     0000* 00249 ;TPA$SUBEXP
                 U.302:   .WORD     <<U.7-U.302>-2>                          ;
 00000000V 0024B ;TPA$ACTION
                 U.303:   .LONG     <<GET_CONT_LINE-U.303>-4>                ;
     0000* 0024F ;TPA$TARGET
                 U.304:   .WORD     <<U.292-U.304>-2>                        ;
     002F  00251 ;TPA$TYPE
                 U.305:   .WORD     47                                       ;
     95F6  00253 ;TPA$TYPE
                 U.306:   .WORD     -27146                                   ;
 00000000V 00255 ;TPA$ACTION
                 U.307:   .LONG     <<MESSAGE_DEFN-U.307>-4>                 ;
     0000* 00259 ;TPA$TARGET
                 U.308:   .WORD     <<U.67-U.308>-2>                         ;
     1DF8  0025B ;TPA$TYPE
                 U.309:   .WORD     7672                                     ;
     0000* 0025D ;TPA$SUBEXP
                 U.310:   .WORD     <<U.298-U.310>-2>                        ;
     0000* 0025F ;TPA$TARGET
                 U.311:   .WORD     <<U.292-U.311>-2>                        ;
           00261 ;DEF_QUAL
                 U.298:   .BLKB     0
     9BF8  00261 ;TPA$TYPE
                 U.312:   .WORD     -25608                                   ;
       01  00263 ;TPA$FLAGS2
                 U.313:   .BYTE     1                                        ;
     0000* 00264 ;TPA$SUBEXP
```

```
                          U.314:   .WORD    <<U.210-U.314>-2>                          ;
00000000' 00266 ;TPA$PARAM
                          U.315:   .ADDRESS P.AAK                                      ;
00000000V 0026A ;TPA$ACTION
                          U.316:   .LONG    <<STORE_NUMBER-U.316>-4>                   ;
     FFFF 0026E ;TPA$TARGET
                          U.317:   .WORD    -1                                         ;
     110D 00270 ;TPA$TYPE
                          U.321:   .WORD    4365                                       ;
    0000* 00272 ;TPA$TARGET
                          U.323:   .WORD    <<U.322-U.323>-2>                          ;
     110E 00274 ;TPA$TYPE
                          U.327:   .WORD    4366                                       ;
    0000* 00276 ;TPA$TARGET
                          U.329:   .WORD    <<J.328-U.329>-2>                          ;
     110F 00278 ;TPA$TYPE
                          U.333:   .WORD    4367                                       ;
    0000* 0027A ;TPA$TARGET
                          U.335:   .WORD    <<U.334-U.335>-2>                          ;
     1110 0027C ;TPA$TYPE
                          U.339:   .WORD    4368                                       ;
    0000* 0027E ;TPA$TARGET
                          U.341:   .WORD    <<U.340-U.341>-2>                          ;
     1511 00280 ;TPA$TYPE
                          U.345:   .WORD    5393                                       ;
    0000* 00282 ;TPA$TARGET
                          U.347:   .WORD    <<U.346-U.347>-2>                          ;
          00284 ;FAOCNT
                          U.322:   .BLKB    0
     003D 00284 ;TPA$TYPE
                          U.349:   .WORD    61                                         ;
     043A 00286 ;TPA$TYPE
                          U.350:   .WORD    1082                                       ;
     9FF8 00288 ;TPA$TYPE
                          U.351:   .WORD    -24584                                     ;
       01 0028A ;TPA$FLAGS2
                          U.352:   .BYTE    1                                          ;
    0000V 0028B ;TPA$SUBEXP
                          U.353:   .WORD    <<U.125-U.353>-2>                          ;
00000000' 0028D ;TPA$PARAM
                          U.354:   .ADDRESS P.AAL                                      ;
00000000V 00291 ;TPA$ACTION
                          U.355:   .LONG    <<STORE_NUMBER-U.355>-4>                   ;
     FFFF 00295 ;TPA$TARGET
                          U.356:   .WORD    -1                                         ;
          00297 ;IDENT
                          U.328:   .BLKB    0
     003D 00297 ;TPA$TYPE
                          U.357:   .WORD    61                                         ;
     043A 00299 ;TPA$TYPE
                          U.358:   .WORD    1082                                       ;
     97F1 0029B ;TPA$TYPE
                          U.359:   .WORD    -26639                                     ;
       01 0029D ;TPA$FLAGS2
                          U.360:   .BYTE    1                                          ;
00000000' 0029E ;TPA$PARAM
                          U.361:   .ADDRESS P.AAM                                      ;
```

```
00000000V 002A2 ;TPA$ACTION
                U.362:  .LONG   <<STORE_STRING-U.362>-4>                ;
     FFFF 002A6 ;TPA$TARGET
                U.363:  .WORD   -1                                      ;
          002A8 ;DETAIL
                U.334:  .BLKB   0
     003D 002A8 ;TPA$TYPE
                U.364:  .WORD   61                                      ;
     043A 002AA ;TPA$TYPE
                U.365:  .WORD   1082                                    ;
     9FF8 002AC ;TPA$TYPE
                U.366:  .WORD   -24584                                  ;
       01 002AE ;TPA$FLAGS2
                U.367:  .BYTE   1                                       ;
    0000V 002AF ;TPA$SUBEXP
                U.368:  .WORD   <<U.125-U.368>-2>                       ;
00000000' 002B1 ;TPA$PARAM
                U.369:  .ADDRESS P.AAN                                  ;
00000000V 002B5 ;TPA$ACTION
                U.370:  .LONG   <<STORE_NUMBER-U.370>-4>                ;
     FFFF 002B9 ;TPA$TARGET
                U.371:  .WORD   -1                                      ;
          002BB ;LANG
                U.340:  .BLKB   0
     003D 002BB ;TPA$TYPE
                U.372:  .WORD   61                                      ;
     043A 002BD ;TPA$TYPE
                U.373:  .WORD   1082                                    ;
     9FF8 002BF ;TPA$TYPE
                U.374:  .WORD   -24584                                  ;
       01 002C1 ;TPA$FLAGS2
                U.375:  .BYTE   1                                       ;
    0000* 002C2 ;TPA$SUBEXP
                U.376:  .WORD   <<U.265-U.376>-2>                       ;
00000000' 002C4 ;TPA$PARAM
                U.377:  .ADDRESS P.AAO                                  ;
00000000V 002C8 ;TPA$ACTION
                U.378:  .LONG   <<STORE_NUMBER-U.378>-4>                ;
     FFFF 002CC ;TPA$TARGET
                U.379:  .WORD   -1                                      ;
          002CE ;USERVAL
                U.346:  .BLKB   0
     003D 002CE ;TPA$TYPE
                U.380:  .WORD   61                                      ;
     043A 002D0 ;TPA$TYPE
                U.381:  .WORD   1082                                    ;
     9FF8 002D2 ;TPA$TYPE
                U.382:  .WORD   -24584                                  ;
       01 002D4 ;TPA$FLAGS2
                U.383:  .BYTE   1                                       ;
    0000V 002D5 ;TPA$SUBEXP
                U.384:  .WORD   <<U.125-U.384>-2>                       ;
00000000' 002D7 ;TPA$PARAM
                U.385:  .ADDRESS P.AAP                                  ;
00000000V 002DB ;TPA$ACTION
                U.386:  .LONG   <<STORE_NUMBER-U.386>-4>                ;
     FFFF 002DF ;TPA$TARGET
```

```
                         U.387:   .WORD    -1                                      ;
                  002E1  ;PARSE_LANG
                         U.265:   .BLKB    0
           9312   002E1  ;TPA$TYPE
                         U.391:   .WORD    -27886                                  ;
             01   002E3  ;TPA$FLAGS2
                         U.392:   .BYTE    1                                       ;
       00000000   002E4  ;TPA$PARAM
                         U.393:   .LONG    0                                       ;
      00000000*   002E8  ;TPA$ACTION
                         U.394:   .LONG    <<SET_NUMBER-U.394>-4>                  ;
           FFFF   002EC  ;TPA$TARGET
                         U.395:   .WORD    -1                                      ;
           9313   002EE  ;TPA$TYPE
                         U.399:   .WORD    -27885                                  ;
             01   002F0  ;TPA$FLAGS2
                         U.400:   .BYTE    1                                       ;
       00000002   002F1  ;TPA$PARAM
                         U.401:   .LONG    2                                       ;
      00000000*   002F5  ;TPA$ACTION
                         U.402:   .LONG    <<SET_NUMBER-U.402>-4>                  ;
           FFFF   002F9  ;TPA$TARGET
                         U.403:   .WORD    -1                                      ;
           9714   002FB  ;TPA$TYPE
                         U.407:   .WORD    -26860                                  ;
             01   002FD  ;TPA$FLAGS2
                         U.408:   .BYTE    1                                       ;
       00000001   002FE  ;TPA$PARAM
                         U.409:   .LONG    1                                       ;
      00000000*   00302  ;TPA$ACTION
                         U.410:   .LONG    <<SET_NUMBER-U.410>-4>                  ;
           FFFF   00306  ;TPA$TARGET
                         U.411:   .WORD    -1                                      ;
                  00308  ;PARSE_SEVERITY
                         U.210:   .BLKB    0
           9315   00308  ;TPA$TYPE
                         U.416:   .WORD    -27883                                  ;
             01   0030A  ;TPA$FLAGS2
                         U.417:   .BYTE    1                                       ;
       00000004   0030B  ;TPA$PARAM
                         U.418:   .LONG    4                                       ;
      00000000*   0030F  ;TPA$ACTION
                         U.419:   .LONG    <<SET_NUMBER-U.419>-4>                  ;
           FFFF   00313  ;TPA$TARGET
                         U.420:   .WORD    -1                                      ;
           9316   00315  ;TPA$TYPE
                         U.424:   .WORD    -27882                                  ;
             01   00317  ;TPA$FLAGS2
                         U.425:   .BYTE    1                                       ;
       00000004   00318  ;TPA$PARAM
                         U.426:   .LONG    4                                       ;
      00000000*   0031C  ;TPA$ACTION
                         U.427:   .LONG    <<SET_NUMBER-U.427>-4>                  ;
           FFFF   00320  ;TPA$TARGET
                         U.428:   .WORD    -1                                      ;
           9317   00322  ;TPA$TYPE
                         U.432:   .WORD    -27881                                  ;
       01  00324 ;TPA$FLAGS2
                 U.433:   .BYTE    1                                            ;
 00000003  00325 ;TPA$PARAM
                 U.434:   .LONG    3                                            ;
 00000000* 00329 ;TPA$ACTION
                 U.435:   .LONG    <<SET_NUMBER-U.435>-4>                       ;
     FFFF  0032D ;TPA$TARGET
                 U.436:   .WORD    -1                                           ;
     9318  0032F ;TPA$TYPE
                 U.440:   .WORD    -27880                                       ;
       01  00331 ;TPA$FLAGS2
                 U.441:   .BYTE    1                                            ;
 00000001  00332 ;TPA$PARAM
                 U.442:   .LONG    1                                            ;
 00000000* 00336 ;TPA$ACTION
                 U.443:   .LONG    <<SET_NUMBER-U.443>-4>                       ;
     FFFF  0033A ;TPA$TARGET
                 U.444:   .WORD    -1                                           ;
     9319  0033C ;TPA$TYPE
                 U.448:   .WORD    -27879                                       ;
       01  0033E ;TPA$FLAGS2
                 U.449:   .BYTE    1                                            ;
 00000002  0033F ;TPA$PARAM
                 U.450:   .LONG    2                                            ;
 00000000* 00343 ;TPA$ACTION
                 U.451:   .LONG    <<SET_NUMBER-U.451>-4>                       ;
     FFFF  00347 ;TPA$TARGET
                 U.452:   .WORD    -1                                           ;
     971A  00349 ;TPA$TYPE
                 U.456:   .WORD    -26854                                       ;
       01  0034B ;TPA$FLAGS2
                 U.457:   .BYTE    1                                            ;
 00000000  0034C ;TPA$PARAM
                 U.458:   .LONG    0                                            ;
 00000000* 00350 ;TPA$ACTION
                 U.459:   .LONG    <<SET_NUMBER-U.459>-4>                       ;
     FFFF  00354 ;TPA$TARGET
                 U.460:   .WORD    -1                                           ;

                          .PSECT   _LIB$KEY0$,NOWRT,  SHR,  PIC,1

           00000 PARSE_KEYS::
                          .BLKB    0
           00000 ;TPA$KEY0
                 U.1:     .BLKB    0
     0000* 00000 ;TPA$KEY
                 U.26:    .WORD    <U.25-U.1>                                   ;
     0000* 00002 ;TPA$KEY
                 U.33:    .WORD    <U.32-U.1>                                   ;
     0000* 00004 ;TPA$KEY
                 U.39:    .WORD    <U.38-U.1>                                   ;
     0000* 00006 ;TPA$KEY
                 U.45:    .WORD    <U.44-U.1>                                   ;
     0000* 00008 ;TPA$KEY
                 U.51:    .WORD    <U.50-U.1>                                   ;
     0000* 0000A ;TPA$KEY
                 U.57:    .WORD    <U.56-U.1>                                   ;
```

; R

```
                          0000* 0000C ;TPA$KEY
                                      U.63:    .WORD   <U.62-U.1>                                    ;
                          0000* 0000E ;TPA$KEY
                                      U.70:    .WORD   <U.69-U.1>                                    ;
                          0000* 00010 ;TPA$KEY
                                      U.76:    .WORD   <U.75-U.1>                                    ;
                          0000* 00012 ;TPA$KEY
                                      U.141:   .WORD   <U.140-U.1>                                   ;
                          0000* 00014 ;TPA$KEY
                                      U.148:   .WORD   <U.147-U.1>                                   ;
                          0000* 00016 ;TPA$KEY
                                      U.155:   .WORD   <U.154-U.1>                                   ;
                          0000* 00018 ;TPA$KEY
                                      U.161:   .WORD   <U.160-U.1>                                   ;
                          0000* 0001A ;TPA$KEY
                                      U.319:   .WORD   <U.318-U.1>                                   ;
                          0000* 0001C ;TPA$KEY
                                      U.325:   .WORD   <U.324-U.1>                                   ;
                          0000* 0001E ;TPA$KEY
                                      U.331:   .WORD   <U.330-U.1>                                   ;
                          0000* 00020 ;TPA$KEY
                                      U.337:   .WORD   <U.336-U.1>                                   ;
                          0000* 00022 ;TPA$KEY
                                      U.343:   .WORD   <U.342-U.1>                                   ;
                          0000* 00024 ;TPA$KEY
                                      U.389:   .WORD   <U.388-U.1>                                   ;
                          0000* 00026 ;TPA$KEY
                                      U.397:   .WORD   <U.396-U.1>                                   ;
                          0000* 00028 ;TPA$KEY
                                      U.405:   .WORD   <U.404-U.1>                                   ;
                          0000* 0002A ;TPA$KEY
                                      U.414:   .WORD   <U.413-U.1>                                   ;
                          0000* 0002C ;TPA$KEY
                                      U.422:   .WORD   <U.421-U.1>                                   ;
                          0000* 0002E ;TPA$KEY
                                      U.430:   .WORD   <U.429-U.1>                                   ;
                          0000* 00030 ;TPA$KEY
                                      U.438:   .WORD   <U.437-U.1>                                   ;
                          0000* 00032 ;TPA$KEY
                                      U.446:   .WORD   <U.445-U.1>                                   ;
                          0000* 00034 ;TPA$KEY
                                      U.454:   .WORD   <U.453-U.1>                                   ;

                                               .PSECT  $PLIT$,NOWRT,NOEXE,2

                 00000003  00000               .LONG   3                                             ;
                 00000000' 00004 P.AAA:        .ADDRESS SYMBOL_NAME                                  ;
       0000001F  00000001  00008               .LONG   1, 31                                         ;
                 00000003  00010               .LONG   3                                             ;
                 00000000' 00014 P.AAB:        .ADDRESS FACILITY_NAME                                ;
       00000009  00000001  00018               .LONG   1, 9                                          ;
                 00000003  00020               .LONG   3                                             ;
                 00000000' 00024 P.AAC:        .ADDRESS FACILITY_NUMBER                              ;
       000007FF  00000000  00028               .LONG   0, 2047                                       ;
                 00000003  00030               .LONG   3                                             ;
                 00000000' 00034 P.AAD:        .ADDRESS DEFAULT_PREFIX                               ;
       00000009  00000001  00038               .LONG   1, 9                                          ;
```

```
                          00000003  00040          .LONG    3                                      ;
                          00000000' 00044 P.AAE:   .ADDRESS MACRO_NAME                             ;
                0000000F  00000001  00048          .LONG    1, 15                                  ;
                          00000001  00050          .LONG    1                                      ;
                          00000000' 00054 P.AAF:   .ADDRESS DEFAULT_SEV                            ;
                          00000003  00058          .LONG    3                                      ;
                          00000000' 0005C P.AAG:   .ADDRESS MESSAGE_NUMBER                         ;
                00000FFF  00000000  00060          .LONG    0, 4095                                ;
                          00000001  00068          .LONG    1                                      ;
                          00000000' 0006C P.AAH:   .ADDRESS LITERAL_VALUE                          ;
                          00000001  00070          .LONG    1                                      ;
                          00000000' 00074 P.AAI:   .ADDRESS LITERAL_VALUE                          ;
                          00000001  00078          .LONG    1                                      ;
                          00000000' 0007C P.AAJ:   .ADDRESS DEFAULT_LANG                           ;
                          00000001  00080          .LONG    1                                      ;
                          00000000' 00084 P.AAK:   .ADDRESS SEVERITY_VALUE                         ;
                          00000003  00088          .LONG    3                                      ;
                          00000000' 0008C P.AAL:   .ADDRESS FAOCNT_VALUE                           ;
                0000001F  00000000  00090          .LONG    0, 31                                  ;
                          00000003  00098          .LONG    3                                      ;
                          00000000' 0009C P.AAM:   .ADDRESS IDENT_VALUE                            ;
                0000000F  00000001  000A0          .LONG    1, 15                                  ;
                          00000003  000A8          .LONG    3                                      ;
                          00000000' 000AC P.AAN:   .ADDRESS DETAIL_VALUE                           ;
                000000FF  00000000  000B0          .LONG    0, 255                                 ;
                          00000001  000B8          .LONG    1                                      ;
                          00000000' 000BC P.AAO:   .ADDRESS LANG_VALUE                             ;
                          00000003  000C0          .LONG    3                                      ;
                          00000000' 000C4 P.AAP:   .ADDRESS USERVAL_VALUE                          ;
                000000FF  00000000  000C8          .LONG    0, 255                                 ;


                                                   .PSECT   $CODE$,NOWRT,2

                               0000 00000 NULL2:   .WORD    Save nothing                           ; 0750
                                 04 00002          RET                                             ;

; Routine Size:  3 bytes,    Routine Base:  $CODE$ + 001A
;  517          0751  1
;  518          0752  1 !
;  519          0753  1 !       Expression evaluation
;  520          0754  1 !
;  521          0755  1
;  522          0756  1 OWN
;  523          0757  1     estack:     VECTOR[50],     ! Expression stack
;  524          0758  1     esp;                        ! Expression stack pointer
;  525          0759  1
;  526          0760  1 MACRO pop = (LOCAL temp; temp=..esp; esp=.esp+4; .temp)%;
;  527          0761  1 MACRO push(value) = (esp=.esp-4; .esp=value)%;
;  528          0762  1 MACRO save = (ap_setup; AP [tpa$l_number] = ..esp; true)%;
;  529          0763  1
;  530          0764  1 ROUTINE init_stack = (esp = estack[50]; true);


                                                              .PSECT  $OWN$,NOEXE,2

                                              00214 ESTACK:  .BLKB   200
                                              002DC ESP:     .BLKB   4


                                                              .PSECT  $CODE$,NOWRT,2

                                         0000 00000 INIT_STACK:
                                                              .WORD   Save nothing                           ; 0764
                      0000'  CF     0000'  CF  9E 00002       MOVAB   ESTACK+200, ESP                        ;
                             50            01  D0 00009       MOVL    #1, R0                                 ;
                                               04 0000C       RET                                            ;

; Routine Size:  13 bytes,    Routine Base:  $CODE$ + 001D


;  531          0765  1 ROUTINE add2 = (.esp = pop + ..esp; save);



                                         0000 00000 ADD2:     .WORD   Save nothing                           ; 0765
                             50     0000'  DF  D0 00002       MOVL    @ESP, TEMP                             ;
                      0000'  CF            04  C0 00007       ADDL2   #4, ESP                                ;
                      0000'  DF            50  C0 0000C       ADDL2   TEMP, @ESP                             ;
                         1C  AC     0000'  DF  D0 00011       MOVL    @ESP, 28(AP)                           ;
                             50            01  D0 00017       MOVL    #1, R0                                 ;
                                               04 0001A       RET                                            ;

; Routine Size:  27 bytes,    Routine Base:  $CODE$ + 002A


;  532          0766  1 ROUTINE sub2 = (LOCAL temp; temp = pop; .esp = ..esp - .temp; save);
```

```
                                   0000 00000 SUB2:    .WORD   Save nothing                                        ; 0766
                  50     0000'  DF   D0 00002          MOVL    @ESP, TEMP                                          ;
           0000'  CF               04   C0 00007       ADDL2   #4, ESP                                             ;
           0000'  DF               50   C2 0000C       SUBL2   TEMP, @ESP                                          ;
              1C  AC     0000'  DF   D0 00011          MOVL    @ESP, 28(AP)                                        ;
                  50            01   D0 00017          MOVL    #1, R0                                              ;
                                     04 0001A          RET                                                         ;

; Routine Size:  27 bytes,    Routine Base:  $CODE$ + 0045


;  533          0767  1 ROUTINE mul2 = (.esp = pop * ..esp; save);



                                   0000 00000 MUL2:    .WORD   Save nothing                                        ; 0767
                  50     0000'  DF   D0 00002          MOVL    @ESP, TEMP                                          ;
           0000'  CF               04   C0 00007       ADDL2   #4, ESP                                             ;
           0000'  DF               50   C4 0000C       MULL2   TEMP, @ESP                                          ;
              1C  AC     0000'  DF   D0 00011          MOVL    @ESP, 28(AP)                                        ;
                  50            01   D0 00017          MOVL    #1, R0                                              ;
                                     04 0001A          RET                                                         ;

; Routine Size:  27 bytes,    Routine Base:  $CODE$ + 0060


;  534          0768  1 ROUTINE div2 = (LOCAL temp; temp = pop; .esp = ..esp / .temp; save);



                                   0000 00000 DIV2:    .WORD   Save nothing                                        ; 0768
                  50     0000'  DF   D0 00002          MOVL    @ESP, TEMP                                          ;
           0000'  CF               04   C0 00007       ADDL2   #4, ESP                                             ;
           0000'  DF               50   C6 0000C       DIVL2   TEMP, @ESP                                          ;
              1C  AC     0000'  DF   D0 00011          MOVL    @ESP, 28(AP)                                        ;
                  50            01   D0 00017          MOVL    #1, R0                                              ;
                                     04 0001A          RET                                                         ;

; Routine Size:  27 bytes,    Routine Base:  $CODE$ + 007B


;  535          0769  1 ROUTINE shift2 = (LOCAL temp; temp = pop; .esp = ..esp ^ .temp; save);



                                   0000 00000 SHIFT2:  .WORD   Save nothing                                        ; 0769
                  50     0000'  DF   D0 00002          MOVL    @ESP, TEMP                                          ;
           0000'  CF               04   C0 00007       ADDL2   #4, ESP                                             ;
                  51               50   D0 0000C       MOVL    TEMP, TEMP                                          ;
                  50     0000'  CF   D0 0000F          MOVL    ESP, R0                                             ;
```

```
                60           60               51  78 00014          ASHL     TEMP, (R0), (R0)                                  ;
                        1C   AC               60  D0 00018          MOVL     (R0), 28(AP)                                      ;
                             50               01  D0 0001C          MOVL     #1, R0                                            ;
                                                  04 0001F          RET                                                        ;

; Routine Size:  32 bytes,    Routine Base:  $CODE$ + 0096


;  536          0770  1 ROUTINE neg1 = (.esp = - ..esp; save);



                                                0000 00000 NEG1:    .WORD    Save nothing                                      ; 0770
                             50      0000'    CF  D0 00002          MOVL     ESP, R0                                           ;
                             60               60  CE 00007          MNEGL    (R0), (R0)                                        ;
                        1C   AC               60  D0 0000A          MOVL     (R0), 28(AP)                                      ;
                             50               01  D0 0000E          MOVL     #1, R0                                            ;
                                                  04 00011          RET                                                        ;

; Routine Size:  18 bytes,    Routine Base:  $CODE$ + 00B6


;  537          0771  1 ROUTINE push_constant = (ap_setup; push(.ap[tpa$l_number]); true);



                                                0000 00000 PUSH_CONSTANT:
                                                                    .WORD    Save nothing                                      ; 0771
                   0000'     CF               04  C2 00002          SUBL2    #4, ESP                                           ;
                   0000'     DF          1C   AC  D0 00007          MOVL     28(AP), @ESP                                      ;
                             50               01  D0 0000D          MOVL     #1, R0                                            ;
                                                  04 00010          RET                                                        ;

; Routine Size:  17 bytes,    Routine Base:  $CODE$ + 00C8


;  538          0772  2 ROUTINE push_symbol = (ap_setup; esp=.esp-4;
;  539          0773  2                 IF lookup_symbol (ap [tpa$l_tokencnt],.esp)
;  540          0774  3                 THEN BEGIN save; RETURN true; END
;  541          0775  1                 ELSE RETURN false;);



                                                0000 00000 PUSH_SYMBOL:
                                                                    .WORD    Save nothing                                      ; 0772
                   0000'     CF               04  C2 00002          SUBL2    #4, ESP                                           ;
                                     0000'    CF  DD 00007          PUSHL    ESP                                               ; 0773
                                        10    AC  9F 0000B          PUSHAB   16(AP)                                            ;
                   0000V     CF               02  FB 0000E          CALLS    #2, LOOKUP_SYMBOL                                 ;
                             0A               50  E9 00013          BLBC     R0, 1$                                            ;
                        1C   AC      0000'    DF  D0 00016          MOVL     @ESP, 28(AP)                                      ; 0774
```

```
                                50              01  D0 0001C            MOVL    #1, R0                                  ; 0775
                                                    04 0001F            RET                                             ;
                                                50  D4 00020 1$:        CLRL    R0                                      ;
                                                    04 00022            RET                                             ;

; Routine Size:  35 bytes,    Routine Base:  $CODE$ + 00D9


;    542          0776  1
;    543        P 0777  1 $state(expression,
;    544          0778  1         ((term)));
;    545        P 0779  1 $state(,
;    546        P 0780  1         ('+',addition),
;    547        P 0781  1         ((subtraction),tpa$_exit),      ! Only if followed by valid expression
;    548        P 0782  1                                         ! to allow for continuation dash (-)
;    549          0783  1         (tpa$_lambda,tpa$_exit));
;    550          0784  1
;    551        P 0785  1 $state(addition,
;    552          0786  1         ((expression),tpa$_exit,add2));
;    553          0787  1
;    554        P 0788  1 $state(subtraction,
;    555          0789  1         ('-'));
;    556        P 0790  1 $state(,
;    557          0791  1         ((expression),tpa$_exit,sub2));
;    558          0792  1
;    559        P 0793  1 $state(term,
;    560          0794  1         ((factor)));
;    561        P 0795  1 $state(,
;    562        P 0796  1         ('@',arith_shift),
;    563        P 0797  1         ('*',multiplication),
;    564        P 0798  1         ((division),tpa$_exit),         ! Only if followed by valid term
;    565        P 0799  1                                         ! to allow for qualifier slash (/)
;    566          0800  1         (tpa$_lambda,tpa$_exit));
;    567          0801  1
;    568        P 0802  1 $state(arith_shift,
;    569          0803  1         ((term),tpa$_exit,shift2));
;    570          0804  1
;    571        P 0805  1 $state(multiplication,
;    572          0806  1         ((term),tpa$_exit,mul2));
;    573          0807  1
;    574        P 0808  1 $state(division,
;    575          0809  1         ('/'));
;    576        P 0810  1 $state(,
;    577          0811  1         ((term),tpa$_exit,div2));
;    578          0812  1
;    579        P 0813  1 $state(factor,
;    580        P 0814  1         ('-',negate),
;    581        P 0815  1         ('(',parens),
;    582        P 0816  1         ((constant),tpa$_exit,push_constant),
;    583          0817  1         (tpa$_symbol,tpa$_exit,push_symbol));
;    584          0818  1
;    585        P 0819  1 $state(negate,
;    586          0820  1         ((factor),tpa$_exit,neg1));
;    587          0821  1
;    588        P 0822  1 $state(parens,
;    589          0823  1         ((expression)));
;    590        P 0824  1 $state(,
:  591          0825  1         (')',tpa$_exit));
:  592          0826  1
:  593        P 0827  1 $state(constant,
:  594        P 0828  1         ('+',constant),
:  595        P 0829  1         ('^',radix),
:  596          0830  1         ((decimal),tpa$_exit));
:  597          0831  1
:  598        P 0832  1 $state(radix,
:  599        F 0833  1         ('O',octal),
:  600        P 0834  1         ('X',hex),
:  601          0835  1         ('D',decimal));
:  602          0836  1
:  603        P 0837  1 $state(octal,
:  604          0838  1         (tpa$_octal,tpa$_exit));
:  605          0839  1
:  606        P 0840  1 $state(hex,
:  607          0841  1         (tpa$_hex,tpa$_exit));
:  608          0842  1
:  609        P 0843  1 $state(decimal,
:  610          0844  1         (tpa$_decimal,tpa$_exit));
:  611          0845  1
:  612          0846  1 ROUTINE null3: NOVALUE =;


                                                        .PSECT  _LIB$STATE$,NOWRT,  SHR,  PIC,1

                                           00356 ;EXPRESSION
                                                 U.125:   .BLKB   0
                                     0DF8  00356 ;TPA$TYPE
                                                 U.462:   .WORD   3576                              ;
                                    0000*  00358 ;TPA$SUBEXP
                                                 U.464:   .WORD   <<U.463-U.464>-2>                 ;
                                     102B  0035A ;TPA$TYPE
                                                 U.465:   .WORD   4139                              ;
                                    0000*  0035C ;TPA$TARGET
                                                 U.467:   .WORD   <<U.466-U.467>-2>                 ;
                                     19F8  0035E ;TPA$TYPE
                                                 U.468:   .WORD   6648                              ;
                                    0000*  00360 ;TPA$SUBEXP
                                                 U.470:   .WORD   <<U.469-U.470>-2>                 ;
                                     FFFF  00362 ;TPA$TARGET
                                                 U.471:   .WORD   -1                                ;
                                     15F6  00364 ;TPA$TYPE
                                                 U.472:   .WORD   5622                              ;
                                     FFFF  00366 ;TPA$TARGET
                                                 U.473:   .WORD   -1                                ;
                                           00368 ;ADDITION
                                                 U.466:   .BLKB   0
                                     9DF8  00368 ;TPA$TYPE
                                                 U.474:   .WORD   -25096                            ;
                                    0000*  0036A ;TPA$SUBEXP
                                                 U.475:   .WORD   <<U.125-U.475>-2>                 ;
                                00000000*  0036C ;TPA$ACTION
                                                 U.476:   .LONG   <<ADD2-U.476>-4>                  ;
                                     FFFF  00370 ;TPA$TARGET
                                                 U.477:   .WORD   -1                                ;
```

```
          00372 ;SUBTRACTION
                U.469:  .BLKB    0
     042D 00372 ;TPA$TYPE
                U.478:  .WORD    1069                          ;
     9DF8 00374 ;TPA$TYPE
                U.479:  .WORD    -25096                        ;
    0000* 00376 ;TPA$SUBEXP
                U.480:  .WORD    <<U.125-U.480>-2>             ;
00000000* 00378 ;TPA$ACTION
                U.481:  .LONG    <<SUB2-U.481>-4>              ;
     FFFF 0037C ;TPA$TARGET
                U.482:  .WORD    -1                            ;
          0037E ;TERM
                U.463:  .BLKB    0
     0DF8 0037E ;TPA$TYPE
                U.483:  .WORD    3576                          ;
    0000* 00380 ;TPA$SUBEXP
                U.485:  .WORD    <<U.484-U.485>-2>             ;
     1040 00382 ;TPA$TYPE
                U.486:  .WORD    4160                          ;
    0000* 00384 ;TPA$TARGET
                U.488:  .WORD    <<U.487-U.488>-2>             ;
     102A 00386 ;TPA$TYPE
                U.489:  .WORD    4138                          ;
    0000* 00388 ;TPA$TARGET
                U.491:  .WORD    <<U.490-U.491>-2>             ;
     19F8 0038A ;TPA$TYPE
                U.492:  .WORD    6648                          ;
    0000* 0038C ;TPA$SUBEXP
                U.494:  .WORD    <<U.493-U.494>-2>             ;
     FFFF 0038E ;TPA$TARGET
                U.495:  .WORD    -1                            ;
     15F6 00390 ;TPA$TYPE
                U.496:  .WORD    5622                          ;
     FFFF 00392 ;TPA$TARGET
                U.497:  .WORD    -1                            ;
          00394 ;ARITH_SHIFT
                U.487:  .BLKB    0
     9DF8 00394 ;TPA$TYPE
                U.498:  .WORD    -25096                        ;
    0000* 00396 ;TPA$SUBEXP
                U.499:  .WORD    <<U.463-U.499>-2>             ;
00000000* 00398 ;TPA$ACTION
                U.500:  .LONG    <<SHIFT2-U.500>-4>            ;
     FFFF 0039C ;TPA$TARGET
                U.501:  .WORD    -1                            ;
          0039E ;MULTIPLICATION
                U.490:  .BLKB    0
     9DF8 0039E ;TPA$TYPE
                U.502:  .WORD    -25096                        ;
    0000* 003A0 ;TPA$SUBEXP
                U.503:  .WORD    <<U.463-U.503>-2>             ;
00000000* 003A2 ;TPA$ACTION
                U.504:  .LONG    <<MUL2-U.504>-4>              ;
     FFFF 003A6 ;TPA$TARGET
                U.505:  .WORD    -1                            ;
          003A8 ;DIVISION
```

```
                          U.493:  .BLKB   0
             042F  003A8  ;TPA$TYPE
                          U.506:  .WORD   1071                                        ;
             9DF8  003AA  ;TPA$TYPE
                          U.507:  .WORD   -25096                                      ;
            0000*  003AC  ;TPA$SUBEXP
                          U.508:  .WORD   <<U.463-U.508>-2>                           ;
        00000000*  003AE  ;TPA$ACTION
                          U.509:  .LONG   <<DIV2-U.509>-4>                            ;
             FFFF  003B2  ;TPA$TARGET
                          U.510:  .WORD   -1                                          ;
                   003B4  ;FACTOR
                          U.484:  .BLKB   0
             102D  003B4  ;TPA$TYPE
                          U.511:  .WORD   4141                                        ;
            0000*  003B6  ;TPA$TARGET
                          U.513:  .WORD   <<U.512-U.513>-2>                           ;
             1028  003B8  ;TPA$TYPE
                          U.514:  .WORD   4136                                        ;
            0000*  003BA  ;TPA$TARGET
                          U.516:  .WORD   <<U.515-U.516>-2>                           ;
             99F8  003BC  ;TPA$TYPE
                          U.517:  .WORD   -26120                                      ;
            0000*  003BE  ;TPA$SUBEXP
                          U.519:  .WORD   <<U.518-U.519>-2>                           ;
        00000000*  003C0  ;TPA$ACTION
                          U.520:  .LONG   <<PUSH_CONSTANT-U.520>-4>                   ;
             FFFF  003C4  ;TPA$TARGET
                          U.521:  .WORD   -1                                          ;
             95F1  003C6  ;TPA$TYPE
                          U.522:  .WORD   -27151                                      ;
        00000000*  003C8  ;TPA$ACTION
                          U.523:  .LONG   <<PUSH_SYMBOL-U.523>-4>                     ;
             FFFF  003CC  ;TPA$TARGET
                          U.524:  .WORD   -1                                          ;
                   003CE  ;NEGATE
                          U.512:  .BLKB   0
             9DF8  003CE  ;TPA$TYPE
                          U.525:  .WORD   -25096                                      ;
            0000*  003D0  ;TPA$SUBEXP
                          U.526:  .WORD   <<U.484-U.526>-2>                           ;
        00000000*  003D2  ;TPA$ACTION
                          U.527:  .LONG   <<NEG1-U.527>-4>                            ;
             FFFF  003D6  ;TPA$TARGET
                          U.528:  .WORD   -1                                          ;
                   003D8  ;PARENS
                          U.515:  .BLKB   0
             0DF8  003D8  ;TPA$TYPE
                          U.529:  .WORD   3576                                        ;
            0000*  003DA  ;TPA$SUBEXP
                          U.530:  .WORD   <<U.125-U.530>-2>                           ;
             1429  003DC  ;TPA$TYPE
                          U.531:  .WORD   5161                                        ;
             FFFF  003DE  ;TPA$TARGET
                          U.532:  .WORD   -1                                          ;
                   003E0  ;CONSTANT
                          U.518:  .BLKB   0
```

```
102B  003E0 ;TPA$TYPE
            U.533:   .WORD    4139                                   ;
0000* 003E2 ;TPA$TARGET
            U.534:   .WORD    <<U.518-U.534>-2>                      ;
105E  003E4 ;TPA$TYPE
            U.535:   .WORD    4190                                   ;
0000* 003E6 ;TPA$TARGET
            U.537:   .WORD    <<U.536-U.537>-2>                      ;
1DF8  003E8 ;TPA$TYPE
            U.538:   .WORD    7672                                   ;
0000* 003EA ;TPA$SUBEXP
            U.540:   .WORD    <<U.539-U.540>-2>                      ;
FFFF  003EC ;TPA$TARGET
            U.541:   .WORD    -1                                     ;
      003EE ;RADIX
            U.536:   .BLKB    0
104F  003EE ;TPA$TYPE
            U.542:   .WORD    4175                                   ;
0000* 003F0 ;TPA$TARGET
            U.544:   .WORD    <<U.543-U.544>-2>                      ;
1058  003F2 ;TPA$TYPE
            U.545:   .WORD    4184                                   ;
0000* 003F4 ;TPA$TARGET
            U.547:   .WORD    <<U.546-U.547>-2>                      ;
1444  003F6 ;TPA$TYPE
            U.548:   .WORD    5188                                   ;
0000* 003F8 ;TPA$TARGET
            U.549:   .WORD    <<U.539-U.549>-2>                      ;
      003FA ;OCTAL
            U.543:   .BLKB    0
15F4  003FA ;TPA$TYPE
            U.550:   .WORD    5620                                   ;
FFFF  003FC ;TPA$TARGET
            U.551:   .WORD    -1                                     ;
      003FE ;HEX
            U.546:   .BLKB    0
15F5  003FE ;TPA$TYPE
            U.552:   .WORD    5621                                   ;
FFFF  00400 ;TPA$TARGET
            U.553:   .WORD    -1                                     ;
      00402 ;DECIMAL
            U.539:   .BLKB    0
15F3  00402 ;TPA$TYPE
            U.554:   .WORD    5619                                   ;
FFFF  00404 ;TPA$TARGET
            U.555:   .WORD    -1                                     ;


                     .PSECT   $CODE$,NOWRT,2

 0000 00000 NULL3:   .WORD    Save nothing                           ; 0846
   04 00002          RET                                             ;

; Routine Size:  3 bytes,    Routine Base:  $CODE$ + 00FC
```

```
1 GLOBAL ROUTINE parse_file =
1
1 !---
1 !
1 !       This routine performs the parsing on the already open
1 !       input file of message definitions.
1 !
1 ! Inputs:
1 !
1 !       None
1 !
1 ! Outputs:
1 !
1 !       Various control blocks describing the definitions.
1 !       (see $MSGDEF)
1 !---
1
2 BEGIN
2
2 LOCAL
2     status;                       ! Status code
2
2 IF .num_files EQL 0               ! First file processed
3 THEN BEGIN
3
3      facility_init();             ! Initialize facility cells
3
3      input_linenum = 0;           ! Zero input line number
3
3      title_text [0] = 19;         ! Length of default title
3      CH$MOVE(.title_text [0], UPLIT('Message definitions'), .title_text [1]);
3
2      END;
2
2 num_files = .num_files + 1;       ! Increment total files parsed
2
2 new_page();                       ! Page eject on each new file
2
2 WHILE get_record()
2 DO
3     BEGIN
3     tparse_block [tpa$l_stringcnt] = .input_record [0];
3     tparse_block [tpa$l_stringptr] = .input_record [1];
3
3     status = lib$tparse(tparse_block,parse_states,parse_keys);
3     IF NOT .status                ! If syntax error detected,
3     THEN
4         BEGIN
4         MAP status: BBLOCK;       ! Get at fields
4         IF NOT .status [sts$v_inhib_msg]        ! If not yet signaled
4         THEN
4             syntax_error(tparse_block,emsg(syntax),tparse_block [tpa$l_tokencnt]);
4         END
3     ELSE
3         IF NOT .line_output       ! If line not yet output,
3         THEN
3             echo_record();        ! then echo the input record
```

```
;  671            0904  2        END;
;  672            0905  2
;  673            0906  2  IF .cli_flags [qual_mdl]          ! Output last buffered line of MDL file
;  674            0907  2  THEN mdl_put_record ( UPLIT (0, UPLIT(0) ), true );
;  675            0908  2
;  676            0909  2  IF .cli_flags [qual_sdl]          ! Output last buffered line of SDL file
;  677            0910  2  THEN
;  678            0911  2      sdl_put_record ( UPLIT (0, UPLIT(0) ), true );
;  679            0912  2
;  680            0913  2  RETURN true;
;  681            0914  2
;  682            0915  1  END;


                                                                          .PSECT  $PLIT$,NOWRT,NOEXE,2

74  69  6E  69  66  65  64  20  65  67  61  73  73  65  4D  000D0 P.AAQ:  .ASCII  \Message definitions\<0>              ;
                                            00  73  6E  6F  69  000DF                                                    ;
                                                  00000000  000E4 P.AAS:  .LONG   0                                    ;
                                                  00000000  000E8 P.AAR:  .LONG   0                                    ;
                                                  00000000' 000EC         .ADDRESS P.AAS                               ;
                                                  00000000  000F0 P.AAU:  .LONG   0                                    ;
                                                  00000000  000F4 P.AAT:  .LONG   0                                    ;
                                                  00000000' 000F8         .ADDRESS P.AAU                               ;


                                                                          .PSECT  $CODE$,NOWRT,2

                                                      007C 00000          .ENTRY  PARSE_FILE, Save R2,R3,R4,R5,R6      ; 0847
                         56       0000'    CF   9E 00002          MOVAB   TPARSE_BLOCK, R6                              ;
                                  0000'    CF   D5 00007          TSTL    NUM_FILES                                     ; 0869
                                           18   12 0000B          BNEQ    1$                                            ;
                  0000V  CF                00   FB 0000D          CALLS   #0, FACILITY_INIT                             ; 0872
                                  0000'    CF   D4 00012          CLRL    INPUT_LINENUM                                 ; 0874
                  0000'  CF                13   D0 00016          MOVL    #19, TITLE_TEXT                               ; 0876
         0000'  DF       0000'  CF   0000'    CF   28 0001B       MOVC3   TITLE_TEXT, P.AAQ, @TITLE_TEXT+4              ; 0877
                                  0000'    CF   D6 00025 1$:      INCL    NUM_FILES                                     ; 0881
                  0000G  CF                00   FB 00029          CALLS   #0, NEW_PAGE                                  ; 0883
                  0000V  CF                00   FB 0002E 2$:      CALLS   #0, GET_RECORD                                ; 0885
                         3F                50   E9 00033          BLBC    R0, 4$                                        ;
                     08  A6       0000'    CF   7D 00036          MOVQ    INPUT_RECORD, TPARSE_BLOCK+8                  ; 0888
                                  0000'    CF   9F 0003C          PUSHAB  PARSE_KEYS                                    ; 0891
                                  0000'    CF   9F 00040          PUSHAB  PARSE_STATES                                  ;
                                           56   DD 00044          PUSHL   R6                                            ;
              00000000G  00                03   FB 00046          CALLS   #3, LIB$TPARSE                                ;
                         52                50   D0 0004D          MOVL    R0, STATUS                                    ;
                         16                52   E8 00050          BLBS    STATUS, 3$                                    ; 0892
            D7           52                1C   E0 00053          BBS     #28, STATUS, 2$                               ; 0896
                                     10    A6   9F 00057          PUSHAB  TPARSE_BLOCK+16                               ; 0898
                              009710FC     8F   DD 0005A          PUSHL   #9900284                                      ;
                                           56   DD 00060          PUSHL   R6                                            ;
                  0000G  CF                03   FB 00062          CALLS   #3, SYNTAX_ERROR                              ;
                                           C5   11 00067          BRB     2$                                            ; 0892
                         C0       020C     C6   E8 00069 3$:      BLBS    LINE_OUTPUT, 2$                               ; 0901
                  0000G  CF                00   FB 0006E          CALLS   #0, ECHO_RECORD                               ; 0903
```

```
                                              B9  11 00073          BRB      2$                                    ; 0885
                  0B       0000G  CF          03  E1 00075 4$:      BBC      #3, CLI_FLAGS, 5$                     ; 0906
                                              01  DD 0007B          PUSHL    #1                                    ; 0907
                                      0000'   CF  9F 0007D          PUSHAB   P.AAR                                 ;
                           0000G  CF          02  FB 00081          CALLS    #2, MDL_PUT_RECORD                    ;
                  0B       0000G  CF          04  E1 00086 5$:      BBC      #4, CLI_FLAGS, 6$                     ; 0909
                                              01  DD 0008C          PUSHL    #1                                    ; 0911
                                      0000'   CF  9F 0008E          PUSHAB   P.AAT                                 ;
                           0000G  CF          02  FB 00092          CALLS    #2, SDL_PUT_RECORD                    ;
                                  50          01  D0 00097 6$:      MOVL     #1, R0                                ; 0913
                                                  04 0009A          RET                                            ; 0915

; Routine Size:  155 bytes,    Routine Base:  $CODE$ + 00FF
```

```
1 ROUTINE get_record =
1
1 !---
1 !
1 !       This routine gets the next input record and upcases
1 !       the record if necessary.
1 !
1 ! Inputs:
1 !
1 !       input_rab = Input RAB block
1 !
1 ! Outputs:
1 !
1 !       input_record = Descriptor of input record
1 !
1 !       r0 = status (already signaled if error)
1 !---
1
2 BEGIN
2
2 OWN
2     buffer:     VECTOR [256,BYTE];      ! Upcased input record
2
2 LOCAL
2     status;
2
2 status = $GET (RAB = input_rab);        ! Get next record
2
2 IF NOT .status                          ! If error detected,
2 THEN
3     BEGIN
3     IF .status NEQ rms$_eof     ! If unexpected GET error,
3     THEN
3         rms_error(emsg(readerr),input_fab,input_rab);
3
3     RETURN .status;                     ! return with status
2     END;
2
2 input_linenum = .input_linenum + 1;     ! Increment input line number
2 line_output = false;                    ! Mark line not yet output
2
2 IF .input_rab [rab$w_rsz] GEQ 1            ! If at least 1 character,
2     AND .(.input_rab [rab$l_rbf])<0,8> EQL form_feed ! and if char = FF,
2 THEN
3     BEGIN
3     new_page();                         ! Cause page eject
3     input_rab [rab$w_rsz] = .input_rab [rab$w_rsz] - 1;
3     input_rab [rab$l_rbf] = .input_rab [rab$l_rbf] + 1;
2     END;
2
2 INCR i FROM 0 TO .input_rab [rab$w_rsz]-1
2 DO
3     BEGIN
3     BIND
3         old = .input_rab [rab$l_rbf]: VECTOR[,BYTE];
3     buffer [.i] = .old [.i];
3     SELECTONEU .old [.i] OF
```

```
;  741          0973  3          SET
;  742          0974  3          ['a' TO 'z']: buffer [.i] = .buffer [.i] + ('A'-'a');
;  743          0975  3
;  744          0976  3          ['<','"']:
;  745          0977  4                  BEGIN
;  746          0978  4                  LOCAL p,len;
;  747          0979  4                  p = CH$FIND_CH(.input_rab [rab$w_rsz]-1-.i, old [.i+1],
;  748          0980  4                          (IF .old [.i] EQL '<' THEN '>' ELSE '"'));
;  749          0981  4                  IF .p NEQ 0                     ! If terminator found,
;  750          0982  4                  THEN
;  751          0983  5                      BEGIN
;  752          0984  5                      len = .p - old [.i];        ! length to skip over
;  753          0985  5                      CH$MOVE(.len, old [.i+1], buffer [.i+1]);
;  754          0986  5                      i = .i + .len;              ! then skip to terminator+1
;  755          0987  4                      END;
;  756          0988  3                  END;
;  757          0989  3          TES;
;  758          0990  2      END;
;  759          0991  2
;  760          0992  2 input_record [0] = .input_rab [rab$w_rsz];
;  761          0993  2 input_record [1] = buffer;
;  762          0994  2
;  763          0995  2 RETURN true;
;  764          0996  2
;  765          0997  1 END;


                                                            .PSECT  $OWN$,NOEXE,2

                                          002E0 BUFFER: .BLKB    256

                                                            .EXTRN  SYS$GET

                                                            .PSECT  $CODE$,NOWRT,2

                                     07FC 00000 GET_RECORD:
                                                            .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10          ; 0916
                   5A      0000'  CF  9E 00002              MOVAB   BUFFER, R10                                ;
                   59      0000G  CF  9E 00007              MOVAB   INPUT_RAB+34, R9                           ;
                              DE  A9  9F 0000C              PUSHAB  INPUT_RAB                                  ; 0942
        00000000G  00             01  FB 0000F              CALLS   #1, SYS$GET                                ;
                   52             50  D0 00016              MOVL    R0, STATUS                                 ;
                   1F             52  E8 00019              BLBS    STATUS, 2$                                 ; 0944
        0001827A   8F             52  D1 0001C              CMPL    STATUS, #98938                             ; 0947
                                  12  13 00023              BEQL    1$                                         ;
                              DE  A9  9F 00025              PUSHAB  INPUT_RAB                                  ; 0949
                           0000G  CF  9F 00028              PUSHAB  INPUT_FAB                                  ;
                        009710B4  8F  DD 0002C              PUSHL   #9900212                                   ;
           0000G   CF             03  FB 00032              CALLS   #3, RMS_ERROR                              ;
                   50             52  D0 00037 1$:          MOVL    STATUS, R0                                 ; 0951
                                      04 0003A              RET                                                ;
                           0000'  CF  D6 0003B 2$:          INCL    INPUT_LINENUM                              ; 0954
                            FF2C  CA  94 0003F              CLRB    LINE_OUTPUT                                ; 0955
                                  69  B5 00043              TSTW    INPUT_RAB+34                               ; 0957
                                  10  13 00045              BEQL    3$                                         ;
                   0C         06  B9  91 00047              CMPB    @INPUT_RAB+40, #12                         ; 0958
```

```
                                 0A  12 0004B        BNEQ    3$                                        ;
            0000G  CF            00  FB 0004D        CALLS   #0, NEW_PAGE                              ; 0961
                   69            B7 00052        DECW    INPUT_RAB+34                              ; 0962
                         06      A9  D6 00054        INCL    INPUT_RAB+40                              ; 0963
                   58            69  3C 00057 3$:    MOVZWL  INPUT_RAB+34, R8                          ; 0966
                   56            01  CE 0005A        MNEGL   #1, I                                     ; 0970
                                 58  11 0005D        BRB     10$                                       ;
      53           56    06      A9  C1 0005F 4$:    ADDL3   INPUT_RAB+40, I, R3                       ; 0971
                 6A46            63  90 00064        MOVB    (R3), BUFFER[I]                           ;
              61   8F            63  91 00068        CMPB    (R3), #97                                 ; 0974
                                 0C  1F 0006C        BLSSU   5$                                        ;
              7A   8F            63  91 0006E        CMPB    (R3), #122                                ;
                                 06  1A 00072        BGTRU   5$                                        ;
                 6A46            20  82 00074        SUBB2   #32, BUFFER[I]                            ;
                                 3D  11 00078        BRB     10$                                       ;
                   22            63  91 0007A 5$:    CMPB    (R3), #34                                 ; 0976
                                 05  13 0007D        BEQL    6$                                        ;
                   3C            63  91 0007F        CMPB    (R3), #60                                 ;
                                 33  12 00082        BNEQ    10$                                       ;
      50           58            56  C3 00084 6$:    SUBL3   I, R8, R0                                 ; 0979
                                 50  D7 00088        DECL    R0                                        ;
      52           56    06      A9  C1 0008A        ADDL3   INPUT_RAB+40, I, R2                       ;
                   3C            63  91 0008F        CMPB    (R3), #60                                 ; 0980
                                 05  12 00092        BNEQ    7$                                        ;
                   51            3E  D0 00094        MOVL    #62, R1                                   ;
                                 03  11 00097        BRB     8$                                        ;
                   51            22  D0 00099 7$:    MOVL    #34, R1                                   ;
   01   A2         50            51  3A 0009C 8$:    LOCC    R1, R0, 1(R2)                             ;
                                 02  12 000A1        BNEQ    9$                                        ;
                                 51  D4 000A3        CLRL    R1                                        ;
                                 51  D5 000A5 9$:    TSTL    P                                         ; 0981
                                 0E  13 000A7        BEQL    10$                                       ;
      57           51            53  C3 000A9        SUBL3   R3, P, LEN                                ; 0984
   01 AA46    01   A2            57  28 000AD        MOVC3   LEN, 1(R2), BUFFER+1[I]                   ; 0985
                   56            57  C0 000B4        ADDL2   LEN, I                                    ; 0986
      A4           56            58  F2 000B7 10$:   AOBLSS  R8, I, 4$                                 ; 0966
            0000'  CF            69  3C 000BB        MOVZWL  INPUT_RAB+34, INPUT_RECORD                ; 0992
            0000'  CF            6A  9E 000C0        MOVAB   BUFFER, INPUT_RECORD+4                    ; 0993
                   50            01  D0 000C5        MOVL    #1, R0                                    ; 0995
                                     04 000C8        RET                                               ; 0997

; Routine Size:  201 bytes,    Routine Base:  $CODE$ + 019A
;  767          0998  1 ROUTINE message_init =
;  768          0999  1
;  769          1000  1 !---
;  770          1001  1 !
;  771          1002  1 !       This routine initializes all the local variables set
;  772          1003  1 !       during parsing of a message definition line so that
;  773          1004  1 !       nothing is taken from a previous definition.
;  774          1005  1 !
;  775          1006  1 ! Inputs:
;  776          1007  1 !
;  777          1008  1 !       See OWN storage in the module header.
;  778          1009  1 !
;  779          1010  1 ! Outputs:
;  780          1011  1 !
;  781          1012  1 !       Same
;  782          1013  1 !---
;  783          1014  1
;  784          1015  2 BEGIN
;  785          1016  2
;  786          1017  2 symbol_name [0] = 0;                  ! Clear length of symbol name
;  787          1018  2 severity_value = -1;                  ! Set to illegal value
;  788          1019  2 lang_value = .default_lang;           ! Set current language default
;  789          1020  2 faocnt_value = 0;                     ! Default is 0
;  790          1021  2 ident_value [0] = 0;                  ! Default is use symbol_name
;  791          1022  2 detail_value = 0;                     ! Default is 0
;  792          1023  2 userval_value = 0;                    ! Default is 0
;  793          1024  2
;  794          1025  2 RETURN true;
;  795          1026  2
;  796          1027  1 END;



                                        0004 00000 MESSAGE_INIT:
                                                           .WORD    Save R2                                    ; 0998
                         52    0000'  CF  9E 00002         MOVAB    SYMBOL_NAME, R2                            ;
                                      62  D4 00007         CLRL     SYMBOL_NAME                                ; 1017
                    08   A2           01  CE 00009         MNEGL    #1, SEVERITY_VALUE                         ; 1018
                    2C   A2     C0    A2  D0 0000D         MOVL     DEFAULT_LANG, LANG_VALUE                   ; 1019
                                0C    A2  D4 00012         CLRL     FAOCNT_VALUE                               ; 1020
                                20    A2  D4 00015         CLRL     IDENT_VALUE                                ; 1021
                                28    A2  D4 00018         CLRL     DETAIL_VALUE                               ; 1022
                                30    A2  D4 0001B         CLRL     USERVAL_VALUE                              ; 1023
                         50           01  D0 0001E         MOVL     #1, R0                                     ; 1025
                                          04 00021         RET                                                 ; 1027

; Routine Size:  34 bytes,    Routine Base:  $CODE$ + 0263
```

```
1 ROUTINE message_defn =
1
1 !---
1 !
1 !       This routine processes the information stored by the
1 !       TPARSE action routines and created the necessary message
1 !       definition blocks to store the data.
1 !
1 ! Inputs:
1 !
1 !       See OWN storage in the module header.
1 !
1 ! Outputs:
1 !
1 !       Control blocks are allocated and linked into the
1 !       message definitions.
1 !---
1
2 BEGIN
2
2 BUILTIN
2     AP,                                 ! Address of tparse block
2     INSQUE;                             ! Insert into linked list
2
2 MAP
2     ap: REF BBLOCK;
2
2 LOCAL
2     code:       REF BBLOCK,             ! Address of CODE block
2     status,
2     msglen;                             ! Length of MSG block
2
2 !
2 !       Check size of global symbol name.  This must be done
2 !       here because it is made up of the 2 separate strings.
2 !
2
2 IF .symbol_name [0] + .default_prefix [0] GTR sym_plus_pre
2 THEN
2     RETURN(syntax_error(.ap,emsg(symtoolng)));  ! then signal it
2
2 !
2 !       Default any unspecified values
2 !
2
2 IF .ident_value [0] EQL 0               ! If /IDENT not specified,
2 THEN
3     BEGIN
3     ident_value [0] = .symbol_name [0]; ! then use symbol name
3     IF .ident_value [0] GTR ident_bufsiz ! If symbol larger than max. ident
3     THEN
3         ident_value [0] = ident_bufsiz; ! then truncate to maximum size
3     CH$MOVE(.ident_value [0], .symbol_name [1], .ident_value [1]);
2     END;
2
2 !
2 !       Allocate the space for the definition
```

; R

```
2 !
2
2 msglen = mrec$c_fixedlen + .ident_value [0] + .message_text [0] + 2;
2 IF .msglen AND 1                              ! If not on word boundary
2 THEN
2     msglen = .msglen + 1;                     ! Force to word boundary
2
2 IF NOT allocate(code$c_length+.msglen,code)     ! Allocate block
2 THEN                                          ! and signal any error
2     RETURN true;                              ! Return no syntax error
2
2 !
2 !       Setup the fields of the CODE/MSG block
2 !
2
3 BEGIN
3
3 LOCAL
3     symbol_buffer: VECTOR [symbol_bufsiz,BYTE], ! Global symbol name
3     symbol_desc: VECTOR [2];                  ! Descriptor of above symbol
3
3 BIND
3     msg = code [code$c_msg,0,0,0]: BBLOCK,          ! MSG block is hung off CODE block
3     msg_code = code [code$l_number]: BBLOCK;        ! To get at STS fields
3
3 code [code$l_number] = 0;                           ! Preset longword
3 msg_code [sts$v_fac_no] = .facility_number;         ! Set facility number
3 msg_code [sts$v_code] = .message_number;            ! Set message number
3
3 IF .severity_value LSS 0                      ! If severity unspecified,
3 THEN
3     IF .default_sev GEQ 0                     ! If default severity specified,
3     THEN
3         severity_value = .default_sev         ! use default severity
3     ELSE
4         BEGIN                                 ! Else,
4         syntax_error(.ap,emsg(nosever));            ! signal unspecified severity level
4         severity_value = sts$k_error;         ! use error to keep going
3         END;
3
3 msg_code [sts$v_severity] = .severity_value;        ! Set severity
3
3 IF NOT .facility_flags [shared_bit]       ! If /SHARED,
3 THEN
3     msg_code [sts$v_fac_sp] = true;           ! then this is facility specific
3
3 CH$COPY(.default_prefix[0], .default_prefix [1],
3         .symbol_name [0], .symbol_name [1],
3         0, sym_plus_pre, symbol_buffer);            ! Copy symbol name
3
3 symbol_desc [0] = .default_prefix [0] + .symbol_name [0]; ! Setup descriptor
3 symbol_desc [1] = symbol_buffer;
3
3 status = add_symbol(symbol_desc, .code [code$l_number]); ! Add to symbol table
3
3 IF NOT .status                                ! If error detected,
3 THEN
```

```
;   912         1142  4       BEGIN
;   913         1143  4       deallocate(code$c_length+.msglen, .code);    ! Send CODE block back
;   914         1144  4       RETURN true;                             ! then return, error already signaled
;   915         1145  3       END;
;   916         1146  3
;   917         1147  3  CH$FILL(0,.msglen,msg);                       ! Zero MSG block
;   918         1148  3  msg [mrec$w_size] = .msglen;                  ! Set length of block
;   919         1149  3  msg [mrec$b_flags] = 0;                       ! Initialize flags
;   920         .150  3  msg [mrec$b_level] = .detail_value;           ! Set detail level value
;   92'         1151  3  msg [mrec$b_faocnt] = .faocnt_value;          ! Set FAO count value
;   922         1152  3  msg [mrec$b_userval] = .userval_value;        ! Set user value
;   923         1153  3  msg [mrec$b_lang] = .lang_value;                      ! Set language number
;   924         1154  3  msg [mrec$b_identlen] = .ident_value [0];! Set ident string (ASCIC)
;   925         1155  3  CH$MOVE(.ident_value[0], .ident_value[1], msg [mrec$t_ident]);
;   926         1156  3  msg [mrec$c_fixedlen + .ident_value [0]+1,0,8,0] =
;   927         1157  3          .message_text [0];                    ! Set message text string (ASCIC)
;   928         1158  3  CH$MOVE(.message_text [0], .message_text [1],
;   929         1159  3          msg [mrec$c_fixedlen+.ident_value[0]+2,0,0,0]);
;   930         1160  2  END;
;   931         1161  2
;   932         1162  2  status = add_message (.code);                 ! Add message to linked list
;   933         1163  2
;   934         1164  2  IF NOT .status                                ! If error detected,
;   935         1165  2  THEN
;   936         1166  3       BEGIN
;   937         1167  3       deallocate(code$c_length+.msglen, .code);    ! Send CODE block back
;   938         1168  3       RETURN true;                             ! return, error already signaled
;   939         1169  2       END;
;   940         1170  2
;   941         1171  2  msg_space = .msg_space + .msglen;             ! Total all space used by MSG blocks
;   942         1172  2  num_messages = .num_messages + 1;             ! Count number of messages in list
;   943         1173  2  message_number = .message_number + 1;         ! Skip to next message number
;   944         1174  2
;   945         1175  2  IF .cli_flags [qual_mdl]                      ! If /MDL specified, then define a constant
;   946         1176  3  THEN BEGIN
;   947         1177  3       mdl_define_constant (symbol_name, code [code$l_number], (NOT literal_flag), .ap);
;   948         1178  3       IF .ap [tpa$l_stringcnt] NEQ 0     ! Still a comment left to parse in this record
;   949         1179  3       THEN new_line = false;             ! Do not start a new line for any comment
;   950         1180  2       END;
;   951         1181  2
;   952         1182  2  IF .cli_flags [qual_sdl]                      ! If /SDL specified, then define a constant
;   953         1183  3  THEN BEGIN
;   954         1184  3       sdl_define_constant (symbol_name, code [code$l_number], (NOT literal_flag), .ap);
;   955         1185  3       IF .ap [tpa$l_stringcnt] NEQ 0     ! Still a comment left to parse in this record
;   956         1186  3       THEN new_line = false;             ! Do not start a new line for any comment
;   957         1187  2       END;
;   958         1188  2
;   959         1189  2  line_with_value (.code [code$l_number]); ! Output line w/msg number
;   960         1190  2  line_output = true;                           ! Mark line already output
;   961         1191  2
;   962         1192  2  RETURN true;
;   963         1193  2
;   964         1194  1  END;


                                                             .EXTRN  MSG$_SYMTOOLNG, MSG$_NOSEVER
```

```
                                           OFFC 00000 MESSAGE_DEFN:
                                                            .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11            ; 1028
                           5E             2C C2 00002       SUBL2    #44, SP                                         ;
                50  0000'  CF     0000'   CF C1 00005       ADDL3    DEFAULT_PREFIX, SYMBOL_NAME, R0                 ; 1065
                           1F             50 D1 0000D       CMPL     R0, #31                                         ;
                                          0E 15 00010       BLEQ     1$                                              ;
                              00000000G   8F DD 00012       PUSHL    #MSG$_SYMTOOLNG                                 ; 1067
                                          5C DD 00018       PUSHL    AP                                              ;
                    0000G  CF             02 FB 0001A       CALLS    #2, SYNTAX_ERROR                                ;
                                             04 0001F       RET                                                      ;
                                  0000'   CF D5 00020 1$:   TSTL     IDENT_VALUE                                     ; 1073
                                           D 12 00024       BNEQ     3$                                              ;
                    0000'  CF     0000'   CF D0 00026       MOVL     SYMBOL_NAME, IDENT_VALUE                        ; 1076
                           0F     0000'   CF D1 0002D       CMPL     IDENT_VALUE, #15                                ; 1077
                                          05 15 00032       BLEQ     2$                                              ;
                    0000'  CF             0F D0 00034       MOVL     #15, IDENT_VALUE                                ; 1079
         0000'  DF  0000'  DF     0000'   CF 28 00039 2$:   MOVC3    IDENT_VALUE, @SYMBOL_NAME+4, @IDENT_VALUE+4     ; 1080
                56  0000'  CF     0000'   CF C1 00043 3$:   ADDL3    MESSAGE_TEXT, IDENT_VALUE, R6                   ; 1087
                           56             0B C0 0004B       ADDL2    #11, MSGLEN                                     ;
                           02             56 E9 0004E       BLBC     MSGLEN, 4$                                      ; 1088
                                          56 D6 00051       INCL     MSGLEN                                          ; 1090
                                          5E DD 00053 4$:   PUSHL    SP                                              ; 1092
                                     08   A6 9F 00055       PUSHAB   8(MSGLEN)                                       ;
                    0000V  CF             02 FB 00058       CALLS    #2, ALLOCATE                                    ;
                           03             50 E8 0005D       BLBS     R0, 5$                                          ;
                                        014D 31 00060       BRW      14$                                             ;
                           59             6E D0 00063 5$:   MOVL     CODE, R9                                        ; 1107
                           57        08   A9 9E 00066       MOVAB    8(R9), R7                                       ;
                           5B        04   A9 9E 0006A       MOVAB    4(R9), R11                                      ; 1108
                                          6B D4 0006E       CLRL     (R11)                                           ; 1110
02   AB         0C         00     0000'   CF F0 00070       INSV     FACILITY_NUMBER, #0, #12, 2(R11)                ; 1111
     6B         0C         03     0000'   CF F0 00078       INSV     MESSAGE_NUMBER, #3, #12, (R11)                  ; 1112
                                  0000'   CF D5 0007F       TSTL     SEVERITY_VALUE                                  ; 1114
                                          20 18 00083       BGEQ     7$                                              ;
                           50     0000'   CF D0 00085       MOVL     DEFAULT_SEV, R0                                 ; 1116
                                          07 19 0008A       BLSS     6$                                              ;
                    0000'  CF             50 D0 0008C       MOVL     R0, SEVERITY_VALUE                              ; 1118
                                          12 11 00091       BRB      7$                                              ;
                              00000000G   8F DD 00093 6$:   PUSHL    #MSG$_NOSEVER                                   ; 1121
                                          5C DD 00099       PUSHL    AP                                              ;
                    0000G  CF             02 FB 0009B       CALLS    #2, SYNTAX_ERROR                                ;
                    0000'  CF             02 D0 000A0       MOVL     #2, SEVERITY_VALUE                              ; 1122
     6B         03         00     0000'   CF F0 000A5 7$:   INSV     SEVERITY_VALUE, #0, #3, (R11)                   ; 1125
                           05     0000'   CF E8 000AC       BLBS     FACILITY_FLAGS, 8$                              ; 1127
                       01  AB        80   8F 88 000B1       BISB2    #128, 1(R11)                                    ; 1129
                           5A             1F D0 000B6 8$:   MOVL     #31, R10                                        ; 1131
                           58        0C   AE 9E 000B9       MOVAB    SYMBOL_BUFFER, R8                               ;
     5A         00  0000'  DF     0000'   CF 2C 000BD       MOVC5    DEFAULT_PREFIX, @DEFAULT_PREFIX+4, #0, -        ;
                                          68    000C6                R10, (R8)                                       ;
                                          14 18 000C7       BGEQ     9$                                              ;
                           58     0000'   CF C0 000C9       ADDL2    DEFAULT_PREFIX, R8                              ;
                           5A     0000'   CF C2 000CE       SUBL2    DEFAULT_PREFIX, R10                             ;
     5A         00  0000'  DF     0000'   CF 2C 000D3       MOVC5    SYMBOL_NAME, @SYMBOL_NAME+4, #0, R10, (R8)      ;
                                          68    000DC                                                                ;
           04   AE  0000'  CF     0000'   CF C1 000DD 9$:   ADDL3    SYMBOL_NAME, DEFAULT_PREFIX, SYMBOL_DESC        ; 1135
                       08  AE        0C   AE 9E 000E6       MOVAB    SYMBOL_BUFFER, SYMBOL_DESC+4                    ; 1136
```

```
                                        6B  DD 000EB            PUSHL    (R11)                                          ; 1138
                                08      AE  9F 000ED            PUSHAB   SYMBOL_DESC                                    ;
                0000V   CF              02  FB 000F0            CALLS    #2, ADD_SYMBOL                                 ;
                        58              50  D0 000F5            MOVL     R0, STATUS                                     ;
                        52              58  E9 000F8            BLBC     STATUS, 10$                                    ; 1140
    56          00      6E              00  2C 000FB            MOVC5    #0, (SP), #0, MSGLEN, (R7)                     ; 1147
                                        67     00100                                                                    ;
                        67              56  B0 00101            MOVW     MSGLEN, (R7)                                   ; 1148
                                03      A7  94 00104            CLRB     3(R7)                                          ; 1149
                04      A7      0000'   CF  90 00107            MOVB     DETAIL_VALUE, 4(R7)                            ; 1150
                05      A7      0000'   CF  90 0010D            MOVB     FAOCNT_VALUE, 5(R7)                            ; 1151
                06      A7      0000'   CF  90 00113            MOVB     USERVAL_VALUE, 6(R7)                           ; 1152
                08      A7      0000'   CF  90 00119            MOVB     LANG_VALUE, 8(R7)                              ; 1153
                        5A      0000'   CF  D0 0011F            MOVL     IDENT_VALUE, R10                               ; 1154
                09      A7              5A  90 00124            MOVB     R10, 9(R7)                                     ;
        0A      A7      0000'   DF      5A  28 00128            MOVC3    R10, @IDENT_VALUE+4, 10(R7)                    ; 1155
                0A AA47         0000'   CF  90 0012F            MOVB     MESSAGE_TEXT, 10(R10)[R7]                      ; 1157
        0B AA47 0000'   DF      0000'   CF  28 00136            MOVC3    MESSAGE_TEXT, @MESSAGE_TEXT+4, 11(R10)[R7]     ; 1159
                                        59  DD 00140            PUSHL    R9                                             ; 1162
                0000V   CF              01  FB 00142            CALLS    #1, ADD_MESSAGE                                ;
                        58              50  D0 00147            MOVL     R0, STATUS                                     ;
                        0C              58  E8 0014A            BLBS     STATUS, 11$                                    ; 1164
                                        59  DD 0014D 10$:       PUSHL    R9                                             ; 1167
                                08      A6  9F 0014F            PUSHAB   8(MSGLEN)                                      ;
                0000V   CF              02  FB 00152            CALLS    #2, DEALLOCATE                                 ;
                                        57  11 00157            BRB      14$                                            ; 1168
                0000'   CF              56  C0 00159 11$:       ADDL2    MSGLEN, MSG_SPACE                              ; 1171
                                0000'   CF  D6 0015E            INCL     NUM_MESSAGES                                   ; 1172
                                0000'   CF  D6 00162            INCL     MESSAGE_NUMBER                                 ; 1173
                19      0000G   CF      03  E1 00166            BBC      #3, CLI_FLAGS, 12$                             ; 1175
                                        5C  DD 0016C            PUSHL    AP                                             ; 1177
                        7E              01  CE 0016E            MNEGL    #1, -(SP)                                      ;
                                        5B  DD 00171            PUSHL    R11                                            ;
                                0000'   CF  9F 00173            PUSHAB   SYMBOL_NAME                                    ;
                0000G   CF              04  FB 00177            CALLS    #4, MDL_DEFINE_CONSTANT                        ;
                                08      AC  D5 0017C            TSTL     8(AP)                                          ; 1178
                                        04  13 0017F            BEQL     12$                                            ;
                                0000'   CF  D4 00181            CLRL     NEW_LINE                                       ; 1179
                19      0000G   CF      04  E1 00185 12$:       BBC      #4, CLI_FLAGS, 13$                             ; 1182
                                        5C  DD 0018B            PUSHL    AP                                             ; 1184
                        7E              01  CE 0018D            MNEGL    #1, -(SP)                                      ;
                                        5B  DD 00190            PUSHL    R11                                            ;
                                0000'   CF  9F 00192            PUSHAB   SYMBOL_NAME                                    ;
                0000G   CF              04  FB 00196            CALLS    #4, SDL_DEFINE_CONSTANT                        ;
                                08      AC  D5 0019B            TSTL     8(AP)                                          ; 1185
                                        04  13 0019E            BEQL     13$                                            ;
                                0000'   CF  D4 001A0            CLRL     NEW_LINE                                       ; 1186
                                        6B  DD 001A4 13$:       PUSHL    (R11)                                          ; 1189
                0000G   CF              01  FB 001A6            CALLS    #1, LINE_WITH_VALUE                            ;
                0000'   CF              01  90 001AB            MOVB     #1, LINE_OUTPUT                                ; 1190
                        50              01  D0 001B0 14$:       MOVL     #1, R0                                         ; 1192
                                            04 001B3            RET                                                     ; 1194

; Routine Size:  436 bytes,    Routine Base:  $CODE$ + 0285
```

```
ROUTINE add_message (code) =

!---
!
!       This routine adds the specified CODE definition block
!       to the linked list.
!
! Inputs:
!
!       code = Address of CODE block
!       tparse_block = Address of TPARSE block
!       message_header = List head of CODE list
!
! Outputs:
!
!       r0 = status (already signaled)
!
!---

BEGIN

MAP
    code:       REF BBLOCK;             ! Address of CODE block

LOCAL
    ptr:        REF BBLOCK,             ! Current position in linked list
    prev:       REF BBLOCK;             ! Previous entry in linked list

prev = message_header;                  ! Start at list head
ptr = .prev [code$l_link];              ! First entry in list

WHILE .ptr NEQ 0                        ! Until we reach end of list,
DO
    BEGIN
    IF (.ptr [code$l_number] AND sts$m_cond_id) GEQU
        (.code [code$l_number] AND sts$m_cond_id)       ! If found position,
    THEN
        EXITLOOP;                       ! then exit the search
    prev = .ptr;                        ! Save address of previous entry
    ptr = .ptr [code$l_link];           ! Skip to next entry
    END;

IF .ptr NEQ 0
    AND ((.ptr [code$l_number] AND sts$m_cond_id) EQL
        (.code [code$l_number] AND sts$m_cond_id))      ! If already there,
THEN
    BEGIN
    BIND msg = ptr [code$c_msg,0,0,0]: BBLOCK;  ! Access msg defn block
    syntax_error(tparse_block,          ! signal dup error
            emsg(dupmsg), .code [code$l_number],
            .msg [mrec$b_identlen], msg [mrec$t_ident]);
    RETURN emsg(dupmsg);
    END;

code [code$l_link] = .prev [code$l_link];       ! Link into list
prev [code$l_link] = .code;
```

```
; 1023          1252  2 RETURN true;
; 1024          1253  2
; 1025          1254  1 END;


                                                                       .EXTRN  MSG$_DUPMSG

                                                   003C 00000 ADD_MESSAGE:
                                                                       .WORD   Save R2,R3,R4,R5                               ; 1195
                            55 00000000G  8F  D0 00002              MOVL    #MSG$_DUPMSG, R5                               ;
                            54     0000'  CF  9E 00009              MOVAB   MESSAGE_HEADER, PREV                           ; 1223
                            50            64  D0 0000E              MOVL    (PREV), PTR                                    ; 1224
                            52        04  AC  D0 00011              MOVL    CODE, R2                                       ; 1230
                                          50  D5 00015 1$:          TSTL    PTR                                            ; 1226
                                          1F  13 00017              BEQL    2$                                             ;
      53          04        A0 F0000007   8F  CB 00019              BICL3   #-268435449, 4(PTR), R3                        ; 1229
      51          04        A2 F0000007   8F  CB 00022              BICL3   #-268435449, 4(R2), R1                         ; 1230
                            51            53  D1 0002B              CMPL    R3, R1                                         ;
                                          08  1E 0002E              BGEQU   2$                                             ;
                            54            50  D0 00030              MOVL    PTR, PREV                                      ; 1233
                            50            60  D0 00033              MOVL    (PTR), PTR                                     ; 1234
                                          DD  11 00036              BRB     1$                                             ; 1226
                                          50  D5 00038 2$:          TSTL    PTR                                            ; 1237
                                          33  13 0003A              BEQL    3$                                             ;
      53          04        A0 F0000007   8F  CB 0003C              BICL3   #-268435449, 4(PTR), R3                        ; 1238
      51          04        A2 F0000007   8F  CB 00045              BICL3   #-268435449, 4(R2), R1                         ; 1239
                            51            53  D1 0004E              CMPL    R3, R1                                         ;
                                          1C  12 00051              BNEQ    3$                                             ;
                            50            08  C0 00053              ADDL2   #8, R0                                         ; 1242
                                      0A  A0  9F 00056              PUSHAB  10(R0)                                         ; 1245
                            7E        09  A0  9A 00059              MOVZBL  9(R0), -(SP)                                   ;
                                      04  A2  DD 0005D              PUSHL   4(R2)                                          ;
                                          55  DD 00060              PUSHL   R5                                             ;
                                   0000'  CF  9F 00062              PUSHAB  TPARSE_BLOCK                                   ; 1243
                  0000G     CF            05  FB 00066              CALLS   #5, SYNTAX_ERROR                               ; 1245
                            50            55  D0 0006B              MOVL    R5, R0                                         ; 1246
                                          04 0006E                  RET                                                    ;
                  04        BC            64  D0 0006F 3$:          MOVL    (PREV), @CODE                                  ; 1249
                            64        04  AC  D0 00073              MOVL    CODE, (PREV)                                   ; 1250
                            50            01  D0 00077              MOVL    #1, R0                                         ; 1252
                                          04 0007A                  RET                                                    ; 1254

; Routine Size:  123 bytes,    Routine Base:  $CODE$ + 0439
; 1027          1255  1 ROUTINE facility_init =
; 1028          1256  1
; 1029          1257  1 !---
; 1030          1258  1 !       This routine initializes the various OWN cells
; 1031          1259  1 !       so that no data is left over from the previous
; 1032          1260  1 !       facility.
; 1033          1261  1 !
; 1034          1262  1 ! Inputs:
; 1035          1263  1 !
; 1036          1264  1 !       See OWN storage in module header.
; 1037          1265  1 !
; 1038          1266  1 ! Outputs:
; 1039          1267  1 !
; 1040          1268  1 !       Same
; 1041          1269  1 !---
; 1042          1270  1
; 1043          1271  2 BEGIN
; 1044          1272  2
; 1045          1273  2 facility_number = 0;                    ! Clear facility number
; 1046          1274  2 facility_name [0] = 0;                  ! and name
; 1047          1275  2 facility_flags = 0;                     ! Clear flags
; 1048          1276  2 message_number = 1;                     ! Start at message 1 (default)
; 1049          1277  2 default_sev = -1;                       ! Mark no severity defined yet
; 1050          1278  2 default_prefix [0] = 0;                 ! No default prefix
; 1051          1279  2
; 1052          1280  2 IF .cli_flags [qual_mdl]
; 1053          1281  3 THEN BEGIN
; 1054          1282  3      mdl_end_struc ();                  ! If /MDL specified, then end structure
; 1055          1283  2      END;
; 1056          1284  2
; 1057          1285  2 IF .cli_flags [qual_sdl]
; 1058          1286  3 THEN BEGIN
; 1059          1287  3      sdl_end_mod ();                    ! If /SDL specified, then end module
; 1060          1288  2      END;
; 1061          1289  2
; 1062          1290  2 RETURN true;
; 1063          1291  2
; 1064          1292  1 END;


                                             0000 00000 FACILITY_INIT:
                                                                  .WORD   Save nothing                             ; 1255
                                  0000'  CF   D4 00002           CLRL    FACILITY_NAME                            ; 1274
                                  0000'  CF   7C 00006           CLRQ    FACILITY_NUMBER                          ; 1273
                   0000'  CF              01  D0 0000A           MOVL    #1, MESSAGE_NUMBER                       ; 1276
                   0000'  CF              01  CE 0000F           MNEGL   #1, DEFAULT_SEV                          ; 1277
                                  0000'  CF   D4 00014           CLRL    DEFAULT_PREFIX                           ; 1278
          05       0000G  CF              03  E1 00018           BBC     #3, CLI_FLAGS, 1$                        ; 1280
                   0000G  CF              00  FB 0001E           CALLS   #0, MDL_END_STRUC                        ; 1282
          05       0000G  CF              04  E1 00023 1$:       BBC     #4, CLI_FLAGS, 2$                        ; 1285
                   0000G  CF              00  FB 00029           CALLS   #0, SDL_END_MOD                          ; 1287
                          50              01  D0 0002E 2$:       MOVL    #1, R0                                   ; 1290
                                              04 00031           RET                                              ; 1292
```

```
; Routine Size:  50 bytes,    Routine Base:  $CODE$ + 04B4
```

```
PAR
V04
```

```
; R
```

```
1 ROUTINE facility_defn =
1
1 !---
1 !
1 !       This routine generates the necessary control blocks
1 !       to describe a newly defined facility.
1 !
1 ! Inputs:
1 !
1 !       See OWN storage in module header.
1 !
1 ! Outputs:
1 !
1 !       The FAC descriptor block is generated.
1 !---
1
2 BEGIN
2
2 BUILTIN
2     AP;
2
2 MAP
2     ap: REF BBLOCK;
2
2 LOCAL
2     status,
2     fac:        REF BBLOCK;             ! Address of FAC block
2
2 !
2 !       Do not allow a facility number other than 0 with /SHARED
2 !
2
2 IF .facility_flags [shared_bit] AND .facility_number NEQ 0
2 THEN
3     BEGIN
3     syntax_error(.ap, emsg(sharconf));  ! Signal syntax error
3     RETURN true;                        ! return no syntax error
2     END;
2
2 !
2 !       Allocate memory for FAC block and fill it
2 !
2
2 IF NOT allocate(fac$c_length,fac)       ! Allocate block
2 THEN                                    ! and signal any error
2     RETURN true;                        ! Return no syntax error
2
2 IF NOT .facility_flags [system_bit]     ! If not /SYSTEM,
2 THEN
2     facility_number = .facility_number
2         OR (sts$m_cust_def ^ -%BITPOSITION(sts$v_fac_no));
2
2 fac [fac$w_number] = .facility_number;  ! Set facility number
2 fac [fac$b_namelen] = .facility_name [0];       ! Set name length
2 CH$MOVE(.facility_name [0], .facility_name [1], ! Set string into block
2         fac [fac$t_name]);
2
```

```
2 IF .macro_name [0] NEQ 0               ! If /MACRO specified
2 THEN facility_flags = .facility_flags OR macro_mask;
2
2 IF .ap [tpa$l_stringcnt] NEQ 0         ! Still a comment left in record
2 THEN new_line = false;                 ! Do not start a new line for a comment
2
2 IF .default_prefix [0] EQL 0           ! If no default prefix specified,
2 THEN
3     BEGIN
3     LOCAL delimdesc:    VECTOR [2];
3
3     delimdesc [0] = 2;
3     delimdesc [1] = UPLIT('$_');       ! Set delimiter to '$_'
3
3     IF .cli_flags [qual_mdl]           ! If /MDL specified, then start structure
4     THEN BEGIN
4          mdl_start_struc (facility_name, .facility_number, macro_name, facility_flags);
3          END;
3
3     IF .cli_flags [qual_sdl]           ! If /SDL specified, then start module
4     THEN BEGIN
4          sdl_start_mod (facility_name, .facility_number, macro_name, facility_flags);
3          END;
3
3     IF NOT .facility_flags [system_bit] ! If user facility,
3     THEN
4         BEGIN
4         delimdesc [0] = .delimdesc [0] - 1; ! Use '_' as delimiter
4         delimdesc [1] = .delimdesc [1] + 1;
3         END;
3
3     default_prefix [0] = .facility_name [0] + .delimdesc [0];
3     default_prefix [1] = prefix_buffer;
3     CH$COPY(.facility_name [0], .facility_name [1], ! Setup default prefix
3         .delimdesc [0], .delimdesc [1],
3         0, prefix_bufsiz, prefix_buffer);
3     END
3
3 ELSE BEGIN
3      facility_flags = .facility_flags OR prefix_mask;
3
3      IF .cli_flags [qual_mdl]           ! If /MDL specified, then start structure
4      THEN BEGIN
4           mdl_start_struc (default_prefix, .facility_number, macro_name, facility_flags );
3           END;
3
3      IF .cli_flags [qual_sdl]           ! If /SDL specified, then start structure
4      THEN BEGIN
4           sdl_start_mod (default_prefix, .facility_number, macro_name, facility_flags );
3           END;
3
2      END;
2
3 BEGIN
3 LOCAL
3     name_buffer:VECTOR [obj$c_symsiz,BYTE],
3     name_desc:  VECTOR [2];
```

```
; 1180          1407  3
; 1181          1408  3 CH$COPY(.fac [fac$b_namelen], fac [fac$t_name], ! Make fac$_FACILITY name
; 1182          1409  3         10, UPLIT('$_FACILITY'),
; 1183          1410  3         0, obj$c_symsiz, name_buffer);
; 1184          1411  3 name_desc [0] = .fac [fac$b_namelen] + 10;      ! Make descriptor of name
; 1185          1412  3 name_desc [1] = name_buffer;
; 1186          1413  3 IF NOT lookup_symbol(name_desc, status) ! If not already in symbol table,
; 1187          1414  3 THEN
; 1188          1415  4     BEGIN
; 1189          1416  4     status = add_symbol(name_desc, .fac [fac$w_number]); ! Add to symbol table
; 1190          1417  4     IF NOT .status                        ! If error detected,
; 1191          1418  4     THEN
; 1192          1419  5         BEGIN
; 1193          1420  5         deallocate(fac$c_length,.fac);  ! Send FAC block back
; 1194          1421  5         RETURN true;                    ! return, error already signaled
; 1195          1422  4         END;
; 1196          1423  3     END;
; 1197          1424  2 END;
; 1198          1425  2
; 1199          1426  2 line_with_value (.fac [fac$w_number]);  ! Output line w/fac number
; 1200          1427  2 line_output = true;                     ! Mark line already output
; 1201          1428  2
; 1202          1429  2 status - add_facility(.fac);            ! Add facility to facility list
; 1203          1430  2
; 1204          1431  2 IF NOT .status                          ! If error detected,
; 1205          1432  2 THEN
; 1206          1433  3     BEGIN
; 1207          1434  3     deallocate(fac$c_length,.fac);      ! Send FAC block back
; 1208          1435  3     RETURN true;
; 1209          1436  2     END;
; 1210          1437  2
; 1211          1438  2 num_facilities = .num_facilities + 1;   ! Increment facilities in list
; 1212          1439  2 fac_space = .fac_space + $BYTEOFFSET(mfac$t_name) + .facility_name [0];
; 1213          1440  2                                         ! Add space to facility table
; 1214          1441  2 RETURN true;
; 1215          1442  2
; 1216          1443  1 END;


                                                                    .PSECT  $PLIT$,NOWRT,NOEXE,2

                                        00  00  5F  24   000FC P.AAV:   .ASCII  \$_\<0><0>                              ;
        00  00  59  54  49  4C  49  43  41  46  5F  24   00100 P.AAW:   .ASCII  \$_FACILITY\<0><0>                      ;

                                                                    .EXTRN  MSG$_SHARCONF

                                                                    .PSECT  $CODE$,NOWRT,2

                                                  0FFC 00000 FACILITY_DEFN:
                                                                    .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11   ; 1293
                            5B      0000'  CF  9E 00002             MOVAB   FACILITY_NAME, R11                      ;
                            5A      0000'  CF  9E 00007             MOVAB   FACILITY_FLAGS, R10                     ;
                            5E             30  C2 0000C             SUBL2   #48, SP                                 ;
                            15             6A  E9 0000F             BLBC    FACILITY_FLAGS, 2$                      ; 1325
                                    FC     AA  D5 00012             TSTL    FACILITY_NUMBER                         ;
                                           10  13 00015             BEQL    2$                                      ;
                                    00000000G  8F  DD 00017          PUSHL   #MSG$_SHARCONF                              ; 1328
                                               5C  DD 0001D          PUSHL   AP                                          ;
                          0000G  CF            02  FB 0001F          CALLS   #2, SYNTAX_ERROR                            ;
                                          015C 31 00024 1$:          BRW     16$                                         ; 1329
                                               5E  DD 00027 2$:      PUSHL   SP                                          ; 1336
                                               16  DD 00029          PUSHL   #22                                         ;
                          0000V  CF            02  FB 0002B          CALLS   #2, ALLOCATE                                ;
                                 F1            50  E9 00030          BLBC    R0, 1$                                      ;
                    04           6A            01  E0 00033          BBS     #1, FACILITY_FLAGS, 3$                      ; 1340
                             FD  AA            08  88 00037          BISB2   #8, FACILITY_NUMBER+1                       ; 1343
                                 56            6E  D0 0003B 3$:      MOVL    FAC, R6                                     ; 1345
                                 57      FC    AA  D0 0003E          MOVL    FACILITY_NUMBER, R7                         ;
                             04  A6            57  B0 00042          MOVW    R7, 4(R6)                                   ;
                             06  A6            6B  90 00046          MOVB    FACILITY_NAME, 6(R6)                        ; 1346
                07      A6   04  BB            6B  28 0004A          MOVC3   FACILITY_NAME, @FACILITY_NAME+4, 7(R6)      ; 1348
                                         50    AA  D5 00050          TSTL    MACRO_NAME                                  ; 1350
                                               03  13 00053          BEQL    4$                                          ;
                                 6A            08  88 00055          BISB2   #8, FACILITY_FLAGS                          ; 1351
                                         08    AC  D5 00058 4$:      TSTL    8(AP)                                       ; 1353
                                               04  13 0005B          BEQL    5$                                          ;
                                       01E8    CA  D4 0005D          CLRL    NEW_LINE                                    ; 1354
                                         30    AA  D5 00061 5$:      TSTL    DEFAULT_PREFIX                              ; 1356
                                               68  12 00064          BNEQ    9$                                          ;
                             28  AE            02  D0 00066          MOVL    #2, DELIMDESC                               ; 1361
                             2C  AE    0000'   CF  9E 0006A          MOVAB   P.AAV, DELIMDESC+4                          ; 1362
                0E        0000G  CF            03  E1 00070          BBC     #3, CLI_FLAGS, 6$                           ; 1364
                                               5A  DD 00076          PUSHL   R10                                         ; 1366
                                         50    AA  9F 00078          PUSHAB  MACRO_NAME                                  ;
                                               57  DD 0007B          PUSHL   R7                                          ;
                                               5B  DD 0007D          PUSHL   R11                                         ;
                          0000G  CF            04  FB 0007F          CALLS   #4, MDL_START_STRUC                         ;
                0F        0000G  CF            04  E1 00084 6$:      BBC     #4, CLI_FLAGS, 7$                           ; 1369
                                               5A  DD 0008A          PUSHL   R10                                         ; 1371
                                         50    AA  9F 0008C          PUSHAB  MACRO_NAME                                  ;
                                         FC    AA  DD 0008F          PUSHL   FACILITY_NUMBER                             ;
                                               5B  DD 00092          PUSHL   R11                                         ;
                          0000G  CF            04  FB 00094          CALLS   #4, SDL_START_MOD                           ;
                06               6A            01  E0 00099 7$:      BBS     #1, FACILITY_FLAGS, 8$                      ; 1374
                                         28    AE  D7 0009D          DECL    DELIMDESC                                   ; 1377
                                         2C    AE  D6 000A0          INCL    DELIMDESC+4                                 ; 1378
                30      AA       6B      28    AE  C1 000A3 8$:      ADDL3   DELIMDESC, FACILITY_NAME, DEFAULT_PREFIX    ; 1381
                             34  AA      10    AA  9E 000A9          MOVAB   PREFIX_BUFFER, DEFAULT_PREFIX+4             ; 1382
                                 58            1F  D0 000AE          MOVL    #31, R8                                     ; 1383
                                 57      10    AA  9E 000B1          MOVAB   PREFIX_BUFFER, R7                           ;
        58          00       04  BB            6B  2C 000B5          MOVC5   FACILITY_NAME, @FACILITY_NAME+4, #0, R8, -  ;
                                               67     000BB                  (R7)                                        ;
                                               3E  18 000BC          BGEQ    11$                                         ;
                                 57            6B  C0 000BE          ADDL2   FACILITY_NAME, R7                           ;
                                 58            6B  C2 000C1          SUBL2   FACILITY_NAME, R8                           ;
        58          00       2C  BE      28    AE  2C 000C4          MOVC5   DELIMDESC, @DELIMDESC+4, #0, R8, (R7)       ;
                                               67     000CB                                                              ;
                                               2E  11 000CC          BRB     11$                                         ; 1356
                                 6A            04  88 000CE 9$:      BISB2   #4, FACILITY_FLAGS                          ; 1389
                0F        0000G  CF            03  E1 000D1          BBC     #3, CLI_FLAGS, 10$                          ; 1391
                                               5A  DD 000D7          PUSHL   R10                                         ; 1393
                                         50    AA  9F 000D9          PUSHAB  MACRO_NAME                                  ;
                                               57  DD 000DC          PUSHL   R7                                          ;
```

```
                                                                M 12
PARSE            PARSE THE MESSAGE SOURCE FILE           16-Sep-1984 02:05:14   VAX-11 Bliss-32 V4.0-742            Page  61
V04-000                                                  14-Sep-1984 12:46:23   DISK$VMSMASTER:[MSGFIL.SRC]PARSE.B32;1   (11)

                                     30    AA  9F 000DE         PUSHAB  DEFAULT_PREFIX                           ;
                      0000G  CF            04  FB 000E1         CALLS   #4, MDL_START_STRUC                      ;
             10       0000G  CF            04  E1 000E6  10$:   BBC     #4, CLI_FLAGS, 11$                       ; 1396
                                           5A  DD 000EC         PUSHL   R10                                      ; 1398
                                     50    AA  9F 000EE         PUSHAB  MACRO_NAME                               ;
                                     FC    AA  DD 000F1         PUSHL   FACILITY_NUMBER                          ;
                                     30    AA  9F 000F4         PUSHAB  DEFAULT_PREFIX                           ;
                      0000G  CF            04  FB 000F7         CALLS   #4, SDL_START_MOD                        ;
                             59      06    A6  9A 000FC  11$:   MOVZBL  6(R6), R9                                ; 1408
                             58            1F  D0 00100         MOVL    #31, R8                                  ;
                             57      10    AE  9E 00103         MOVAB   NAME_BUFFER, R7                          ;
     58      00          07  A6            59  2C 00107         MOVC5   R9, 7(R6), #0, R8, (R7)                  ;
                                           67     0010D                                                          ;
                                           0E  18 0010E         BGEQ    12$                                      ;
                             57            59  C0 00110         ADDL2   R9, R7                                   ;
                             58            59  C2 00113         SUBL2   R9, R8                                   ;
     58      00       0000'  CF            0A  2C 00116         MOVC5   #10, P.AAW, #0, R8, (R7)                 ;
                                           67     0011D                                                          ;
                         08  AE      06    A6  9A 0011E  12$:   MOVZBL  6(R6), NAME_DESC                         ; 1411
                         08  AE            0A  C0 00123         ADDL2   #10, NAME_DESC                           ;
                         0C  AE      10    AE  9E 00127         MOVAB   NAME_BUFFER, NAME_DESC+4                 ; 1412
                                     04    AE  9F 0012C         PUSHAB  STATUS                                   ; 1413
                                     0C    AE  9F 0012F         PUSHAB  NAME_DESC                                ;
                      0000V  CF            02  FB 00132         CALLS   #2, LOOKUP_SYMBOL                        ;
                             14            50  E8 00137         BLBS    R0, 13$                                  ;
                             7E      04    A6  3C 0013A         MOVZWL  4(R6), -(SP)                             ; 1416
                                     0C    AE  9F 0013E         PUSHAB  NAME_DESC                                ;
                      0000V  CF            02  FB 00141         CALLS   #2, ADD_SYMBOL                           ;
                         04  AE            50  D0 00146         MOVL    R0, STATUS                               ;
                             1D      04    AE  E9 0014A         BLBC    STATUS, 14$                              ; 1417
                             7E      04    A6  3C 0014E  13$:   MOVZWL  4(R6), -(SP)                             ; 1426
                      0000G  CF            01  FB 00152         CALLS   #1, LINE_WITH_VALUE                      ;
                       01E4  CA            01  90 00157         MOVB    #1, LINE_OUTPUT                          ; 1427
                                           56  DD 0015C         PUSHL   R6                                       ; 1429
                      0000V  CF            01  FB 0015E         CALLS   #1, ADD_FACILITY                         ;
                         04  AE            50  D0 00163         MOVL    R0, STATUS                               ;
                             0B      04    AE  E8 00167         BLBS    STATUS, 15$                              ; 1431
                                           56  DD 0016B  14$:   PUSHL   R6                                       ; 1434
                                           16  DD 0016D         PUSHL   #22                                      ;
                      0000V  CF            02  FB 0016F         CALLS   #2, DEALLOCATE                           ;
                                           0D  11 00174         BRB     16$                                      ; 1435
                                     1C    AB  D6 00176  15$:   INCL    NUM_FACILITIES                           ; 1438
             50          20  AB            6B  C1 00179         ADDL3   FACILITY_NAME, FAC_SPACE, R0             ; 1439
                         20  AB      03    A0  9E 0017E         MOVAB   3(R0), FAC_SPACE                         ;
                             50            01  D0 00183  16$:   MOVL    #1, R0                                   ; 1441
                                               04 00186         RET                                              ; 1443

; Routine Size:  391 bytes,    Routine Base:  $CODE$ + 04E6
```

```
1 ROUTINE add_facility (fac) =
1
1 !---
1 !
1 !       This routine adds a specified facility definition
1 !       block (FAC) to the defined facility list.
1 !
1 ! Inputs:
1 !
1 !       fac = Address of FAC block
1 !       tparse_block = Address of TPARSE block
1 !       facility_header = List head for defined facilities
1 !
1 ! Outputs:
1 !
1 !       r0 = status (already signaled)
1 !
1 !---
1
2 BEGIN
2
2 MAP
2     fac:        REF BBLOCK;             ! Address of FAC block
2
2 LOCAL
2     ptr:        REF BBLOCK,             ! Current position in linked list
2     prev:       REF BBLOCK;             ! Previous entry in linked list
2
2 prev = facility_header;                 ! Start at list head
2 ptr = .prev [fac$l_link];               ! First entry in list
2
2 WHILE .ptr NEQ 0                        ! Until we reach end of list
2 DO
3     BEGIN
3     IF .ptr [fac$w_number] GEQU .fac [fac$w_number]     ! If found position,
3     THEN
3         EXITLOOP;                       ! then exit the search
3     prev = .ptr;                        ! Save address of previous entry
3     ptr = .ptr [fac$l_link];            ! Skip to next entry
2     END;
2
2 IF .ptr NEQ 0
2     AND .ptr [fac$w_number] EQL .fac [fac$w_number]     ! If already defined,
2 THEN
3     BEGIN
3     IF CH$NEQ(.fac [fac$b_namelen], fac [fac$t_name],
3             .ptr [fac$b_namelen], ptr [fac$t_name], 0)
3                                         ! and the facility names are different
3         AND .fac [fac$w_number] NEQ 0   ! excluding facility number 0
3     THEN
3         syntax_error(tparse_block,      ! signal facility conflict error
3                 emsg(conffac),
3                 .fac [fac$w_number],
3                 .ptr [fac$b_namelen],ptr [fac$t_name]);
3     RETURN emsg(conffac);
2     END;
2
```

```
; 1275          1501  2 fac [fac$l_link] = .prev [fac$l_link];  . Link into facility list
; 1276          1502  2 prev [fac$l_link] = .fac;
; 1277          1503  2
; 1278          1504  2 RETURN true;
; 1279          1505  2
; 1280          1506  1 END;



                                                                     .EXTRN  MSG$_CONFFAC

                                          00FC 00000 ADD_FACILITY:
                                                                     .WORD   Save R2,R3,R4,R5,R6,R7                  ; 1444
                            57 00000000G  8F  D0 00002               MOVL    #MSG$_CONFFAC, R7                       ;
                            56      0000'  CF  9E 00009               MOVAB   FACILITY_HEADER, PREV                   ; 1472
                            54             66  D0 0000E               MOVL    (PREV), PTR                             ; 1473
                            55         04  AC  D0 00011               MOVL    FAC, R5                                 ; 1478
                                           54  D5 00015 1$:           TSTL    PTR                                     ; 1475
                                           0F  13 00017               BEQL    2$                                      ;
                      04    A5         04  A4  B1 00019               CMPW    4(PTR), 4(R5)                           ; 1478
                                           08  1E 0001E               BGEQU   2$                                      ;
                            56             54  D0 00020               MOVL    PTR, PREV                               ; 1481
                            54             64  D0 00023               MOVL    (PTR), PTR                              ; 1482
                                           ED  11 00026               BRB     1$                                      ; 1475
                                           54  D5 00028 2$:           TSTL    PTR                                     ; 1485
                                           38  13 0002A               BEQL    4$                                      ;
                      04    A5         04  A4  B1 0002C               CMPW    4(PTR), 4(R5)                           ; 1486
                                           31  12 00031               BNEQ    4$                                      ;
                            51         06  A5  9A 00033               MOVZBL  6(R5), R1                               ; 1489
                            50         06  A4  9A 00037               MOVZBL  6(PTR), R0                              ; 1490
        50           00     07    A5          51  2D 0003B            CMPC5   R1, 7(R5), #0, R0, 7(PTR)               ;
                                       07  A4     00041                                                              ;
                                           1B  13 00043               BEQL    3$                                      ;
                                       04  A5  B5 00045               TSTW    4(R5)                                   ; 1492
                                           16  13 00048               BEQL    3$                                      ;
                                       07  A4  9F 0004A               PUSHAB  7(PTR)                                  ; 1497
                            7E         06  A4  9A 0004D               MOVZBL  6(PTR), -(SP)                           ;
                            7E         04  A5  3C 00051               MOVZWL  4(R5), -(SP)                            ;
                                           57  DD 00055               PUSHL   R7                                      ;
                                     0000'  CF  9F 00057               PUSHAB  TPARSE_BLOCK                            ; 1494
                    0000G   CF             05  FB 0005B               CALLS   #5, SYNTAX_ERROR                        ; 1497
                            50             57  D0 00060 3$:           MOVL    R7, R0                                  ; 1498
                                               04 00063               RET                                             ;
                            65             66  D0 00064 4$:           MOVL    (PREV), (R5)                            ; 1501
                            66             55  D0 00067               MOVL    R5, (PREV)                              ; 1502
                            50             01  D0 0006A               MOVL    #1, R0                                  ; 1504
                                               04 0006D               RET                                             ; 1506

; Routine Size:  110 bytes,    Routine Base:  $CODE$ + 066D
```

```
1 ROUTINE add_symbol (name_desc, value) =
1
1 !---
1 !
1 !       This routine adds a given symbol name and value to
1 !       the symbol table.
1 !
1 ! Inputs:
1 !
1 !       name_desc = Address of descriptor of symbol name
1 !       value = Value to be assigned to the symbol
1 !
1 ! Outputs:
1 !
1 !       r0 = status (already signaled)
1 !---
1
2 BEGIN
2
2 MAP
2     name_desc:  REF VECTOR;               ! Address of name descriptor
2
2 LOCAL
2     entry:      REF BBLOCK,               ! Address of symbol table entry
2     status;
2
2 IF lookup_symbol (.name_desc, status)    ! If already in symbl table,
2 THEN
3     BEGIN
3     syntax_error(tparse_block,emsg(dupsym));
3     RETURN emsg(dupsym);                  ! return duplicate symbol
2     END;
2
2 IF .name_desc [0] GTRU obj$c_symsiz      ! If symbol length GTR max,
2 THEN
2     name_desc [0] = obj$c_symsiz;         ! then truncate it
2
2 status = allocate(sym$c_length,entry);   ! Allocate a symbol entry
2
2 IF NOT .status                            ! If could not allocate storage,
2 THEN
3     BEGIN
3     syntax_error(tparse_block,.status);
3     RETURN .status;                       ! return with status (already signaled)
2     END;
2
2 entry [sym$l_value] = .value;            ! Set value of symbol
2 entry [sym$b_symlen] = .name_desc [0];   ! Set length of symbol
2 CH$MOVE(.name_desc [0], .name_desc [1], entry [sym$t_symbol]);
2
2 entry [sym$l_link] = .symbol_header;     ! Link into front of symbol table list
2 symbol_header = .entry;
2
2 RETURN true;
2
1 END;
```

```
                                                        .EXTRN  MSG$_DUPSYM

                                   00FC 00000 ADD_SYMBOL:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7                          ; 1507
                  57 00000000G  8F  D0 00002            MOVL    #MSG$_DUPSYM, R7                                ;
                  5E            08  C2 00009            SUBL2   #8, SP                                          ;
                                5E  DD 0000C            PUSHL   SP                                              ; 1533
                  52        04  AC  D0 0000E            MOVL    NAME_DESC, R2                                   ;
                                52  DD 00012            PUSHL   R2                                              ;
            0000V CF            02  FB 00014            CALLS   #2, LOOKUP_SYMBOL                               ;
                  0F            50  E9 00019            BLBC    R0, 1$                                          ;
                                57  DD 0001C            PUSHL   R7                                              ; 1536
                         0000'  CF  9F 0001E            PUSHAB  TPARSE_BLOCK                                    ;
            0000G CF            02  FB 00022            CALLS   #2, SYNTAX_ERROR                                ;
                  50            57  D0 00027            MOVL    R7, R0                                          ; 1537
                                    04 0002A            RET                                                     ;
                  1F            62  D1 0002B 1$:        CMPL    (R2), #31                                       ; 1540
                                03  1B 0002E            BLEQU   2$                                              ;
                  62            1F  D0 00030            MOVL    #31, (R2)                                       ; 1542
                            04  AE  9F 00033 2$:        PUSHAB  ENTRY                                           ; 1544
                                28  DD 00036            PUSHL   #40                                             ;
            0000V CF            02  FB 00038            CALLS   #2, ALLOCATE                                    ;
                  6E            50  D0 0003D            MOVL    R0, STATUS                                      ;
                  0F            6E  E8 00040            BLBS    STATUS, 3$                                      ; 1546
                                6E  DD 00043            PUSHL   STATUS                                          ; 1549
                         0000'  CF  9F 00045            PUSHAB  TPARSE_BLOCK                                    ;
            0000G CF            02  FB 00049            CALLS   #2, SYNTAX_ERROR                                ;
                  50            6E  D0 0004E            MOVL    STATUS, R0                                      ; 1550
                                    04 00051            RET                                                     ;
                  56        04  AE  D0 00052 3$:        MOVL    ENTRY, R6                                       ; 1553
               04 A6        08  AC  D0 00056            MOVL    VALUE, 4(R6)                                    ;
               08 A6            62  90 0005B            MOVB    (R2), 8(R6)                                     ; 1554
      09   A6  04 B2            62  28 0005F            MOVC3   (R2), @4(R2), 9(R6)                             ; 1555
                  66     0000'  CF  D0 00065            MOVL    SYMBOL_HEADER, (R6)                             ; 1557
            0000' CF            56  D0 0006A            MOVL    R6, SYMBOL_HEADER                               ; 1558
                  50            01  D0 0006F            MOVL    #1, R0                                          ; 1560
                                    04 00072            RET                                                     ; 1562

; Routine Size:  115 bytes,    Routine Base:  $CODE$ + 06DB
; 1339          1563  1 ROUTINE lookup_symbol (name_desc, value) =
; 1340          1564  1
; 1341          1565  1 !---
; 1342          1566  1 !
; 1343          1567  1 !       This routine looks up a given symbol in the symbol
; 1344          1568  1 !       table and returns the value associated with it.
; 1345          1569  1 !
; 1346          1570  1 ! Inputs:
; 1347          1571  1 !
; 1348          1572  1 !       name_desc = Descriptor of desired symbol name
; 1349          1573  1 !       value = Address of longword to receive value if found
; 1350          1574  1 !
; 1351          1575  1 ! Outputs:
; 1352          1576  1 !
; 1353          1577  1 !       value = Value of symbol if found
; 1354          1578  1 !       r0 = status
; 1355          1579  1 !---
; 1356          1580  1
; 1357          1581  2 BEGIN
; 1358          1582  2
; 1359          1583  2 MAP
; 1360          1584  2     name_desc:  REF VECTOR;             ! Address of descriptor
; 1361          1585  2
; 1362          1586  2 LOCAL
; 1363          1587  2     ptr:        REF BBLOCK;             ! Pointer into list
; 1364          1588  2
; 1365          1589  2 ptr = .symbol_header;                  ! Start at first entry
; 1366          1590  2
; 1367          1591  2 WHILE .ptr NEQ 0                       ! Until end of list
; 1368          1592  2 DO
; 1369          1593  3     BEGIN
; 1370          1594  3     IF CH$EQL(.ptr [sym$b_symlen], ptr [sym$t_symbol],
; 1371          1595  3             .name_desc [0], .name_desc [1])
; 1372          1596  3     THEN
; 1373          1597  4         BEGIN
; 1374          1598  4         .value = .ptr [sym$l_value];    ! Return value of symbol
; 1375          1599  4         RETURN true;                    ! and exit successful
; 1376          1600  3         END;
; 1377          1601  3     ptr = .ptr [sym$l_link];            ! If no match, go to next entry
; 1378          1602  2     END;
; 1379          1603  2
; 1380          1604  2 RETURN false;                          ! return symbol not f)und
; 1381          1605  2
; 1382          1606  1 END;



                                                  003C 00000 LOOKUP_SYMBOL:
                                                                 .WORD   Save R2,R3,R4,R5                          ; 1563
                              54     0000'  CF  D0 00002         MOVL    SYMBOL_HEADER, PTR                        ; 1589
                              55        04  AC  D0 00007         MOVL    NAME_DESC, R5                             ; 1595
                                            54  D5 0000B 1$:     TSTL    PTR                                       ; 1591
                                            1D  13 0000D         BEQL    3$                                        ;
                              50        08  A4  9A 0000F         MOVZBL  8(PTR), R0                                ; 1594
    04   BC           00        09      A4      50  2D 00013         CMPC5   R0, 9(PTR), #0, @NAME_DESC, @4(R5)        ;
```

```
                        04   B5      0001A                                                         ;
                             09  12  0001C           BNEQ    2$                                    ;
              08   BC   04   A4  D0  0001E           MOVL    4(PTR), @VALUE                        ; 1598
                   50        01  D0  00023           MOVL    #1, R0                                ; 1599
                                 04  00026           RET                                           ;
                   54        64  D0  00027 2$:       MOVL    (PTR), PTR                            ; 1601
                             DF  11  0002A           BRB     1$                                    ; 1591
                             50  D4  0002C 3$:       CLRL    R0                                    ; 1604
                                 04  0002E           RET                                           ; 1606

; Routine Size:  47 bytes,    Routine Base:  $CODE$ + 074E
; 1384          1607  1 ROUTINE find_eos =
; 1385          1608  1
; 1386          1609  1 !---
; 1387          1610  1 !
; 1388          1611  1 !       This action routine finds the end of the message text string.
; 1389          1612  1 !       It uses the first character of the token (tpa$l_char) as the
; 1390          1613  1 !       string terminator - other routines may place a specific
; 1391          1614  1 !       terminator in tpa$l_char and call this routine.  The descriptor
; 1392          1615  1 !       of the message text is stored away.
; 1393          1616  1 !
; 1394          1617  1 ! Inputs:
; 1395          1618  1 !
; 1396          1619  1 !       tpa$l_char = String terminator
; 1397          1620  1 !
; 1398          1621  1 ! Outputs:
; 1399          1622  1 !
; 1400          1623  1 !       message_text = Descriptor of actual message text
; 1401          1624  1 !---
; 1402          1625  1
; 1403          1626  2 BEGIN
; 1404          1627  2
; 1405          1628  2 BUILTIN AP,CALLG;
; 1406          1629  2 MAP ap: REF BBLOCK;
; 1407          1630  2
; 1408          1631  2 LOCAL p;                          ! Temporary string pointer
; 1409          1632  2
; 1410          1633  2 p = CH$FIND_CH(.ap [tpa$l_stringcnt], .ap [tpa$l_stringptr], .ap [tpa$l_char]);
; 1411          1634  2 IF .p EQL 0                       ! If terminator not found,
; 1412          1635  2 THEN
; 1413          1636  2     RETURN false;                 ! then return syntax error
; 1414          1637  2
; 1415          1638  2 ap [tpa$l_tokencnt] = .p - .ap [tpa$l_stringptr];
; 1416          1639  2 ap [tpa$l_tokenptr] = .ap [tpa$l_stringptr];
; 1417          1640  2 ap [tpa$l_stringcnt] = .ap [tpa$l_stringcnt] - (.ap [tpa$l_tokencnt]+1);
; 1418          1641  2 ap [tpa$l_stringptr] = .ap [tpa$l_tokenptr] + (.ap [tpa$l_tokencnt]+1);
; 1419          1642  2
; 1420          1643  2 ap [tpa$l_param] = PLIT(message_text,0,message_bufsiz);
; 1421          1644  2 CALLG(.ap, store_string)         ! Call store string with tparse_block
; 1422          1645  2
; 1423          1646  1 END;


                                                                   .PSECT  $PLIT$,NOWRT,NOEXE,2

                                          00000003  0010C          .LONG   3                                              ;
                                          00000000' 00110 P.AAX:   .ADDRESS MESSAGE_TEXT                                  ;
                                00000100  00000000  00114          .LONG   0, 256                                         ;


                                                                   .PSECT  $CODE$,NOWRT,2

                                               0000 00000 FIND_EOS:
                                                                   .WORD   Save nothing                                   ; 1607
                   0C    BC      08    AC      18    AC  3A 00002  LOCC    24(AP), 8(AP), @12(AP)                        ; 1633
                                                     02  12 00009  BNEQ    1$                                             ;
```

```
                                   51  D4 0000B          CLRL    R1                              ;
                                   51  D5 0000D 1$:      TSTL    P                               ; 1634
                                   03  12 0000F          BNEQ    2$                              ;
                                   50  D4 00011          CLRL    R0                              ; 1636
                                       04 00013          RET                                     ;
       10   AC          51     0C  AC  C3 00014 2$:      SUBL3   12(AP), P, 16(AP)               ; 1638
                  14    AC     0C  AC  D0 0001A          MOVL    12(AP), 20(AP)                  ; 1639
            50    08    AC     10  AC  C3 0001F          SUBL3   16(AP), 8(AP), R0               ; 1640
                  08    AC     FF  A0  9E 00025          MOVAB   -1(R0), 8(AP)                   ;
            50    14    AC     10  AC  C1 0002A          ADDL3   16(AP), 20(AP), R0              ; 1641
                  0C    AC     01  A0  9E 00030          MOVAB   1(R0), 12(AP)                   ;
                  20    AC  0000'  CF  9E 00035          MOVAB   P.AAX, 32(AP)                   ; 1643
               0000V    CF         6C  FA 0003B          CALLG   (AP), STORE_STRING              ; 1644
                                       04 00040          RET                                     ; 1646

; Routine Size:  65 bytes,    Routine Base:  $CODE$ + 077D
```

```
: 1425          1647  1 ROUTINE find_endvers =
: 1426          1648  1
: 1427          1649  1 !++
: 1428          1650  1 !
: 1429          1651  1 !       This routine finds the end of the version (.IDENT) text string.
: 1430          1652  1 !       It uses the first character of the token (tpa$l_char) as the
: 1431          1653  1 !       string terminator.  The descriptor of the ident string is stored.
: 1432          1654  1 !
: 1433          1655  1 !  Inputs:
: 1434          1656  1 !
: 1435          1657  1 !       tpa$l_char -- string terminator
: 1436          1658  1 !
: 1437          1659  1 !  Outputs:
: 1438          1660  1 !
: 1439          1661  1 !       version_num -- descriptor of version string
: 1440          1662  1 !
: 1441          1663  1 !--
: 1442          1664  1
: 1443          1665  2 BEGIN
: 1444          1666  2
: 1445          1667  2 BUILTIN AP, CALLG;
: 1446          1668  2 MAP ap: REF BBLOCK;
: 1447          1669  2
: 1448          1670  2 LOCAL p;                          !temporary string pointer
: 1449          1671  2
: 1450          1672  2 p = CH$FIND_CH(.ap[tpa$l_stringcnt], .ap[tpa$l_stringptr], .ap[tpa$l_char]);
: 1451          1673  2
: 1452          1674  2 IF .p EQL 0                       ! If terminator not found...
: 1453          1675  2 THEN
: 1454          1676  2     RETURN FALSE;                 !  then signal syntax error
: 1455          1677  2
: 1456          1678  2 ap [tpa$l_tokencnt] = .p - .ap [tpa$l_stringptr];
: 1457          1679  2 ap [tpa$l_tokenptr] = .ap [tpa$l_stringptr];
: 1458          1680  2 ap [tpa$l_stringcnt] = .ap [tpa$l_stringcnt] - (.ap [tpa$l_tokencnt]+1);
: 1459          1681  2 ap [tpa$l_stringptr] = .ap [tpa$l_tokenptr] + (.ap [tpa$l_tokencnt]+1);
: 1460          1682  2
: 1461          1683  2 ap [tpa$l_param] = PLIT(version_num,0,obj$c_symsiz);
: 1462          1684  2 CALLG(.ap, store_string)        ! Call store string with tparse_block
: 1463          1685  2
: 1464          1686  1 END;


                                                                   .PSECT  $PLIT$,NOWRT,NOEXE,2

                                       00000003  0011C             .LONG   3                                        :
                                       00000000' 00120 P.AAY:      .ADDRESS VERSION_NUM                             :
                              0000001F 00000000  00124             .LONG   0, 31                                    :


                                                                   .PSECT  $CODE$,NOWRT,2

                                            0000 00000 FIND_ENDVERS:
                                                                   .WORD   Save nothing                             : 1647
                 0C    BC       08   AC      18   AC  3A 00002     LOCC    24(AP), 8(AP), @12(AP)                   : 1672
                                                  02  12 00009     BNEQ    1$                                       :
```

```
                                             51  D4 0000B        CLRL    R1                                  :
                                             51  D5 0000D 1$:    TSTL    P                                   : 1674
                                             03  12 0000F        BNEQ    2$                                  :
                                             50  D4 00011        CLRL    R0                                  : 1676
                                                 04 00013        RET                                         :
         10   AC           51       0C       AC  C3 00014 2$:    SUBL3   12(AP), P, 16(AP)                   : 1678
                     14    AC       0C       AC  D0 0001A        MOVL    12(AP), 20(AP)                      : 1679
              50     08    AC       10       AC  C3 0001F        SUBL3   16(AP), 8(AP), R0                   : 1680
                     08    AC       FF       A0  9E 00025        MOVAB   -1(R0), 8(AP)                       :
              50     14    AC       10       AC  C1 0002A        ADDL3   16(AP), 20(AP), R0                  : 1681
                     0C    AC       01       A0  9E 00030        MOVAB   1(R0), 12(AP)                       :
                     20    AC    0000'       CF  9E 00035        MOVAB   P.AAY, 32(AP)                       : 1683
                  0000V    CF                6C  FA 0003B        CALLG   (AP), STORE_STRING                  : 1684
                                                 04 00040        RET                                         : 1686

; Routine Size:  65 bytes,    Routine Base:  $CODE$ + 07BE
; 1466          1687  1 ROUTINE get_cont_line =
; 1467          1688  1
; 1468          1689  1 !---
; 1469          1690  1 !
; 1470          1691  1 !       This routine is called as an action routine if
; 1471          1692  1 !       a dash (-) is the last token on the line before
; 1472          1693  1 !       a comment or end of line.  The next record is
; 1473          1694  1 !       retrieved and the tparse block is updated so that
; 1474          1695  1 !       parsing continues with the continuation line.
; 1475          1696  1 !
; 1476          1697  1 ! Inputs:
; 1477          1698  1 !
; 1478          1699  1 !       ap = tparse block
; 1479          1700  1 !
; 1480          1701  1 ! Outputs:
; 1481          1702  1 !
; 1482          1703  1 !       tparse block is updated
; 1483          1704  1 !---
; 1484          1705  1
; 1485          1706  2 BEGIN
; 1486          1707  2
; 1487          1708  2 BUILTIN
; 1488          1709  2     AP;                         ! Address of tparse block
; 1489          1710  2
; 1490          1711  2 MAP
; 1491          1712  2     AP:         REF BBLOCK;     ! Address as structure
; 1492          1713  2
; 1493          1714  2 LOCAL
; 1494          1715  2     count,                      ! Count of characters passed over
; 1495          1716  2     status;                     ! status code
; 1496          1717  2
; 1497          1718  2 IF NOT .line_output            ! If line not yet output,
; 1498          1719  2 THEN
; 1499          1720  2     echo_record();              ! then echo the input record
; 1500          1721  2
; 1501          1722  2 status = get_record();         ! Get next record from input stream
; 1502          1723  2 IF NOT .status                 ! If error detected,
; 1503          1724  2 THEN
; 1504          1725  2     RETURN .status;             ! return with status
; 1505          1726  2
; 1506          1727  2 ap [tpa$l_stringcnt] = .input_record [0];
; 1507          1728  2 ap [tpa$l_stringptr] = .input_record [1];
; 1508          1729  2
; 1509          1730  2 RETURN true;
; 1510          1731  2
; 1511          1732  1 END;



                                                0000 00000 GET_CONT_LINE:
                                                                .WORD   Save nothing                                    ; 1687
                        05      0000'   CF   E8 00002           BLBS    LINE_OUTPUT, 1$                                 ; 1718
                0000G   CF              00   FB 00007           CALLS   #0, ECHO_RECORD                                 ; 1720
                F98A    CF              00   FB 0000C 1$:       CALLS   #0, GET_RECORD                                  ; 1722
                        09              50   E9 00011           BLBC    STATUS, 2$                                      ; 1723
```

```
                    08  AC      0000'  CF  7D 00014         MOVQ    INPUT_RECORD, 8(AP)                        ; 1727
                        50             01  D0 0001A         MOVL    #1, R0                                     ; 1730
                                           04 0001D 2$:     RET                                                ; 1732

; Routine Size:  30 bytes,    Routine Base:  $CODE$ + 07FF
```

```
1 ROUTINE define_literal =
1
1 !---
1 !
1 !       This routine adds a user specified literal to the
1 !       symbol table to be output in the object module.
1 !
1 ! Inputs:
1 !
1 !       literal_name = Descriptor of symbol name
1 !       literal_value = Value to assign to the symbol
1 !
1 ! Outputs:
1 !
1 !       literal_value has been automatically incremented by one
1 !               to provide default for next literal parameter.
1 !---
1
2 BEGIN
2
2 BUILTIN
2     AP;                                 ! Address of tparse block
2
2 LOCAL
2     status;
2
2 MAP
2     ap: REF BBLOCK;
2
2 IF NOT add_symbol (literal_name, .literal_value) ! add to symbol table
2 THEN
2     RETURN true;                        ! return, error already signaled
2
2 IF .cli_flags [qual_mdl]                ! If /MDL specified, then define a constant
3 THEN BEGIN
3      status = mdl_define_constant (literal_name, literal_value, literal_flag, tparse_block);
3      IF .ap [tpa$l_stringcnt] NEQ 0     ! Still a comment left in record
3      THEN
3         IF NOT .status          ! If error on return, don't append  comment
3         THEN
3             new_line = true
3         ELSE                    ! else do append comment
3             new_line = false;
2      END;
2
2 IF .cli_flags [qual_sdl]                ! If /SDL specified, then define a constant
3 THEN BEGIN
3      status = sdl_define_constant (literal_name, literal_value, literal_flag, tparse_block);
3      IF .ap [tpa$l_stringcnt] NEQ 0     ! Still a comment left in record
3      THEN
3         IF NOT .status          ! If error on return, don't append comment
3         THEN
3             new_line = true
3         ELSE                    ! else do append comment
3             new_line = false;
2      END;
2
```

```
; 1570          1790  2 literal_value = .literal_value + 1;     . Autoincrement for next symbol
; 1571          1791  2
; 1572          1792  2 RETURN true;
; 1573          1793  2
; 1574          1794  1 END;



                                                     0004 00000 DEFINE_LITERAL:
                                                                        .WORD   Save R2                                     ; 1733
                              52       0000'   CF    9E 00002           MOVAB   LITERAL_VALUE, R2
                                               62    DD 00007           PUSHL   LITERAL_VALUE                               ; 1762
                                          F8   A2    9F 00009           PUSHAB  LITERAL_NAME
                        FEAD    CF             02    FB 0000C           CALLS   #2, ADD_SYMBOL
                                50             50    E9 00011           BLBC    R0, 5$
                  21    0000G   CF             03    E1 00014           BBC     #3, CLI_FLAGS, 2$                           ; 1766
                                        FDF8   C2    9F 0001A           PUSHAB  TPARSE_BLOCK                                ; 1768
                                               7E    D4 0001E           CLRL    -(SP)
                                               52    DD 00020           PUSHL   R2
                                          F8   A2    9F 00022           PUSHAB  LITERAL_NAME
                        0000G   CF             04    FB 00025           CALLS   #4, MDL_DEFINE_CONSTANT
                                          08   AC    D5 0002A           TSTL    8(AP)                                       ; 1769
                                               0C    13 0002D           BEQL    2$
                              06               50    E8 0002F           BLBS    STATUS, 1$                                  ; 1771
                         08   A2               01    D0 00032           MOVL    #1, NEW_LINE                                ; 1773
                                               03    11 00036           BRB     2$
                                          08   A2    D4 00038 1$:       CLRL    NEW_LINE                                    ; 1775
                  21    0000G   CF             04    E1 0003B 2$:       BBC     #4, CLI_FLAGS, 4$                           ; 1778
                                        FDF8   C2    9F 00041           PUSHAB  TPARSE_BLOCK                                ; 1780
                                               7E    D4 00045           CLRL    -(SP)
                                               52    DD 00047           PUSHL   R2
                                          F8   A2    9F 00049           PUSHAB  LITERAL_NAME
                        0000G   CF             04    FB 0004C           CALLS   #4, SDL_DEFINE_CONSTANT
                                          08   AC    D5 00051           TSTL    8(AP)                                       ; 1781
                                               0C    13 00054           BEQL    4$
                              06               50    E8 00056           BLBS    STATUS, 3$                                  ; 1783
                         08   A2               01    D0 00059           MOVL    #1, NEW_LINE                                ; 1785
                                               03    11 0005D           BRB     4$
                                          08   A2    D4 0005F 3$:       CLRL    NEW_LINE                                    ; 1787
                                               62    D6 00062 4$:       INCL    LITERAL_VALUE                               ; 1790
                              50               01    D0 00064 5$:       MOVL    #1, R0                                      ; 1792
                                                     04 00067           RET                                                 ; 1794

; Routine Size:  104 bytes,    Routine Base:  $CODE$ + 081D
; 1576          1795  1 ROUTINE set_title =
; 1577          1796  1
; 1578          1797  1 !---
; 1579          1798  1 !
; 1580          1799  1 !       This routine saves the string from the current position
; 1581          1800  1 !       to the end of the line as the listing title.
; 1582          1801  1 !
; 1583          1802  1 ! Inputs:
; 1584          1803  1 !
; 1585          1804  1 !       tpa$l_tokenptr = Address of start of title string
; 1586          1805  1 !
; 1587          1806  1 ! Outputs:
; 1588          1807  1 !
; 1589          1808  1 !       title_text = Descriptor of title text
; 1590          1809  1 !---
; 1591          1810  1
; 1592          1811  2 BEGIN
; 1593          1812  2
; 1594          1813  2 BUILTIN
; 1595          1814  2     AP,CALLG;
; 1596          1815  2
; 1597          1816  2 MAP
; 1598          1817  2     AP:         REF BBLOCK;             ! Address of tparse block
; 1599          1818  2
; 1600          1819  2 ap [tpa$l_tokencnt] = .input_record [1] + .input_record [0] -
; 1601          1820  2             .ap [tpa$l_tokenptr];
; 1602          1821  2 ap [tpa$l_stringcnt] = 0;               ! Gobble rest of line
; 1603          1822  2
; 1604          1823  2 ap [tpa$l_tokenptr] = .input_rab[rab$l_rbf] + ! Maintain case of title as entered.
; 1605          1824  2             .ap[tpa$l_tokenptr] - .input_record [1];
; 1606          1825  2
; 1607          1826  2 ap [tpa$l_param] = PLIT(title_text,0,title_bufsiz); ! Place to store text
; 1608          1827  2 CALLG(.ap,store_string)                 ! Call store string and save it
; 1609          1828  2
; 1610          1829  1 END;


                                                          .PSECT  $PLIT$,NOWRT,NOEXE,2

                          00000003  0012C             .LONG   3                                                    ;
                          00000000' 00130 P.AAZ:      .ADDRESS TITLE_TEXT                                         ;
                 00000080 00000000  00134             .LONG   0, 128                                               ;


                                                          .PSECT  $CODE$,NOWRT,2

                                0000 00000 SET_TITLE:
                                                          .WORD   Save nothing                                 ; 1795
         50     0000'  CF     0000'  CF  C1 00002         ADDL3   INPUT_RECORD, INPUT_RECORD+4, R0             ; 1819
    10   AC            50       14   AC  C3 0000A         SUBL3   20(AP), R0, 16(AP)                           ; 1820
                                08   AC  D4 00010         CLRL    8(AP)                                        ; 1821
         50     0000G  CF       14   AC  C1 00013         ADDL3   20(AP), INPUT_RAB+40, R0                     ; 1824
    14   AC            50     0000'  CF  C3 0001A         SUBL3   INPUT_RECORD+4, R0, 20(AP)                   ;
                20     AC     0000'  CF  9E 00021         MOVAB   P.AAZ, 32(AP)                                ; 1826
              0000V    CF            6C  FA 00027         CALLG   (AP), STORE_STRING                           ; 1827
```

```
                                                  04 0002C        RET                                                        ; 1829

; Routine Size:  45 bytes,    Routine Base:  $CODE$ + 0885
; 1612          1830  1 ROUTINE set_module =
; 1613          1831  1
; 1614          1832  1 !---
; 1615          1833  1 !
; 1616          1834  1 !       This routine saves the current token as the module name.
; 1617          1835  1 !
; 1618          1836  1 ! Inputs:
; 1619          1837  1 !
; 1620          1838  1 !       tpa$l_tokenptr = Descriptor of module name
; 1621          1839  1 !
; 1622          1840  1 ! Outputs:
; 1623          1841  1 !
; 1624          1842  1 !       module_name = Descriptor of module name
; 1625          1843  1 !---
; 1626          1844  1
; 1627          1845  2 BEGIN
; 1628          1846  2
; 1629          1847  2 BUILTIN AP,CALLG;
; 1630          1848  2
; 1631          1849  2 MAP AP:         REF BBLOCK;             ! Address of tparse block
; 1632          1850  2
; 1633          1851  2 module_name [1] = module_buffer;        ! Set address of buffer
; 1634          1852  2 ap [tpa$l_param] = PLIT(module_name,1,obj$c_symsiz); ! Place to store text
; 1635          1853  2 CALLG(.ap,store_string)                 ! Call store string and save it
; 1636          1854  2
; 1637          1855  1 END;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                          00000003  0013C       .LONG   3                                     :
                                          00000000G 00140 P.ABA: .ADDRESS MODULE_NAME                         :
                                0000001F  00000001  00144       .LONG   1, 31                                 :


                                                                .PSECT  $CODE$,NOWRT,2

                                               0000 00000 SET_MODULE:
                                                                .WORD   Save nothing                          : 1830
                       0000G  CF     0000'  CF  9E 00002        MOVAB   MODULE_BUFFER, MODULE_NAME+4          : 1851
                          20  AC     0000'  CF  9E 00009        MOVAB   P.ABA, 32(AP)                         : 1852
                       0000V  CF            6C  FA 0000F        CALLG   (AP), STORE_STRING                    : 1853
                                                04 00014        RET                                           : 1855

; Routine Size:  21 bytes,    Routine Base:  $CODE$ + 08B2
; 1639          1856  1 ROUTINE build_version =
; 1640          1857  1
; 1641          1858  1 !---
; 1642          1859  1 !
; 1643          1860  1 !       This routine builds a version or ident string for the
; 1644          1861  1 !               message object file.
; 1645          1862  1 !
; 1646          1863  1 ! Inputs:
; 1647          1864  1 !
; 1648          1865  1 !       tpa$l_tokenptr = Pointer to current version character
; 1649          1866  1 !
; 1650          1867  1 ! Outputs:
; 1651          1868  1 !
; 1652          1869  1 !       version_num = Descriptor of version string
; 1653          1870  1 !---
; 1654          1871  1
; 1655          1872  2 BEGIN
; 1656          1873  2
; 1657          1874  2 BUILTIN
; 1658          1875  2     AP,CALLG;
; 1659          1876  2
; 1660          1877  2 MAP
; 1661          1878  2     ap:         REF BBLOCK;             ! Address of tparse block
; 1662          1879  2
; 1663          1880  2 version_num [1] = version_buffer;
; 1664          1881  2 ap[tpa$l_param] = PLIT (version_num,0,obj$c_symsiz);
; 1665          1882  2 CALLG(.ap,store_string);
; 1666          1883  2
; 1667          1884  2 RETURN true;
; 1668          1885  2
; 1669          1886  1 END;


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                    00000003  0014C             .LONG   3                                       ;
                                    00000000' 00150 P.ABB:      .ADDRESS VERSION_NUM                            ;
                          0000001F  00000000  00154             .LONG   0, 31                                   ;


                                                                .PSECT  $CODE$,NOWRT,2

                                         0000 00000 BUILD_VERSION:
                                                                .WORD   Save nothing                            ; 1856
                  0000'  CF      0000'  CF  9E 00002            MOVAB   VERSION_BUFFER, VERSION_NUM+4           ; 1880
                    20   AC      0000'  CF  9E 00009            MOVAB   P.ABB, 32(AP)                           ; 1881
                  0000V  CF             6C  FA 0000F            CALLG   (AP), STORE_STRING                      ; 1882
                         50             01  D0 00014            MOVL    #1, R0                                  ; 1884
                                            04 00017            RET                                             ; 1886

; Routine Size:  24 bytes,    Routine Base:  $CODE$ + 08C7
; 1671        1887  1 ROUTINE store_number =
; 1672        1888  1
; 1673        1889  1 !---
; 1674        1890  1 !
; 1675        1891  1 !       This routine stores the numeric value of a qualifier
; 1676        1892  1 !       into a specified location as long as it is within
; 1677        1893  1 !       certain limits.  If not, a syntax error is signaled.
; 1678        1894  1 !
; 1679        1895  1 ! Inputs:
; 1680        1896  1 !
; 1681        1897  1 !       tpa$l_number = Number to be stored
; 1682        1898  1 !       tpa$l_param = Address of a 3-longword argument list:
; 1683        1899  1 !                       1) Address of longword to receive value
; 1684        1900  1 !                       2) Minimum legal value (unsigned)
; 1685        1901  1 !                       3) Maximum legal value (unsigned)
; 1686        1902  1 !               If 2nd and 3rd arguments are not specified, no
; 1687        1903  1 !                       checking will be done.
; 1688        1904  1 !
; 1689        1905  1 ! Outputs:
; 1690        1906  1 !
; 1691        1907  1 !       Either the number is stored or an error is signaled.
; 1692        1908  1 !---
; 1693        1909  1
; 1694        1910  2 BEGIN
; 1695        1911  2
; 1696        1912  2 BUILTIN
; 1697        1913  2     AP;                               ! Register AP
; 1698        1914  2
; 1699        1915  2 MAP
; 1700        1916  2     AP:         REF BBLOCK;           ! Define TPARSE block format
; 1701        1917  2
; 1702        1918  2 LOCAL
; 1703        1919  2     args:       REF VECTOR,           ! Address of argument list
; 1704        1920  2     value;                            ! Value of expression
; 1705        1921  2
; 1706        1922  2 args = .ap [tpa$l_param]-4;           ! Get address of arguments
; 1707        1923  2                                       ! (PLIT value is addr of FIRST arg)
; 1708        1924  2
; 1709        1925  2 value = .ap [tpa$l_number];           ! Get value
; 1710        1926  2 IF .args [0] GEQ 2                    ! If 2nd, 3rd args specified,
; 1711        1927  2 THEN IF .value LSSU .args [2]         ! If less than minimum
; 1712        1928  2     OR .value GTRU .args [3]          ! or greater than maximum,
; 1713        1929  2 THEN
; 1714        1930  2     RETURN(syntax_error(.ap,emsg(badvalue)));   ! then signal illegal value
; 1715        1931  2
; 1716        1932  2 .args [1] = .value;                   ! Store value into longword
; 1717        1933  2
; 1718        1934  2 RETURN true;
; 1719        1935  2
; 1720        1936  1 END;


                                                        .EXTRN  MSG$_BADVALUE

                                    000C 00000 STORE_NUMBER:
                                                        .WORD   Save R2,R3                          ; 1887
```

```
                     52        20   AC               04  C3 00002             SUBL3    #4, 32(AP), ARGS                                ; 1922
                                    53        1C     AC  D0 00007             MOVL     28(AP), VALUE                                   ; 1925
                                    02               62  D1 0000B             CMPL     (ARGS), #2                                      ; 1926
                                                     1A  19 0000E             BLSS     2$                                              ;
                               08   A2               53  D1 00010             CMPL     VALUE, 8(ARGS)                                  ; 1927
                                                     06  1F 00014             BLSSU    1$                                              ;
                               0C   A2               53  D1 00016             CMPL     VALUE, 12(ARGS)                                 ; 1928
                                                     0E  1B 0001A             BLEQU    2$                                              ;
                                         00000000G   8F  DD 0001C  1$:        PUSHL    #MSG$_BADVALUE                                  ; 1930
                                                     5C  DD 00022             PUSHL    AP                                              ;
                            0000G   CF               02  FB 00024             CALLS    #2, SYNTAX_ERROR                                ;
                                                         04 00029             RET                                                      ;
                               04   B2               53  D0 0002A  2$:        MOVL     VALUE, @4(ARGS)                                 ; 1932
                                    50               01  D0 0002E             MOVL     #1, R0                                          ; 1934
                                                         04 00031             RET                                                      ; 1936

; Routine Size:  50 bytes,    Routine Base:  $CODE$ + 08DF
```

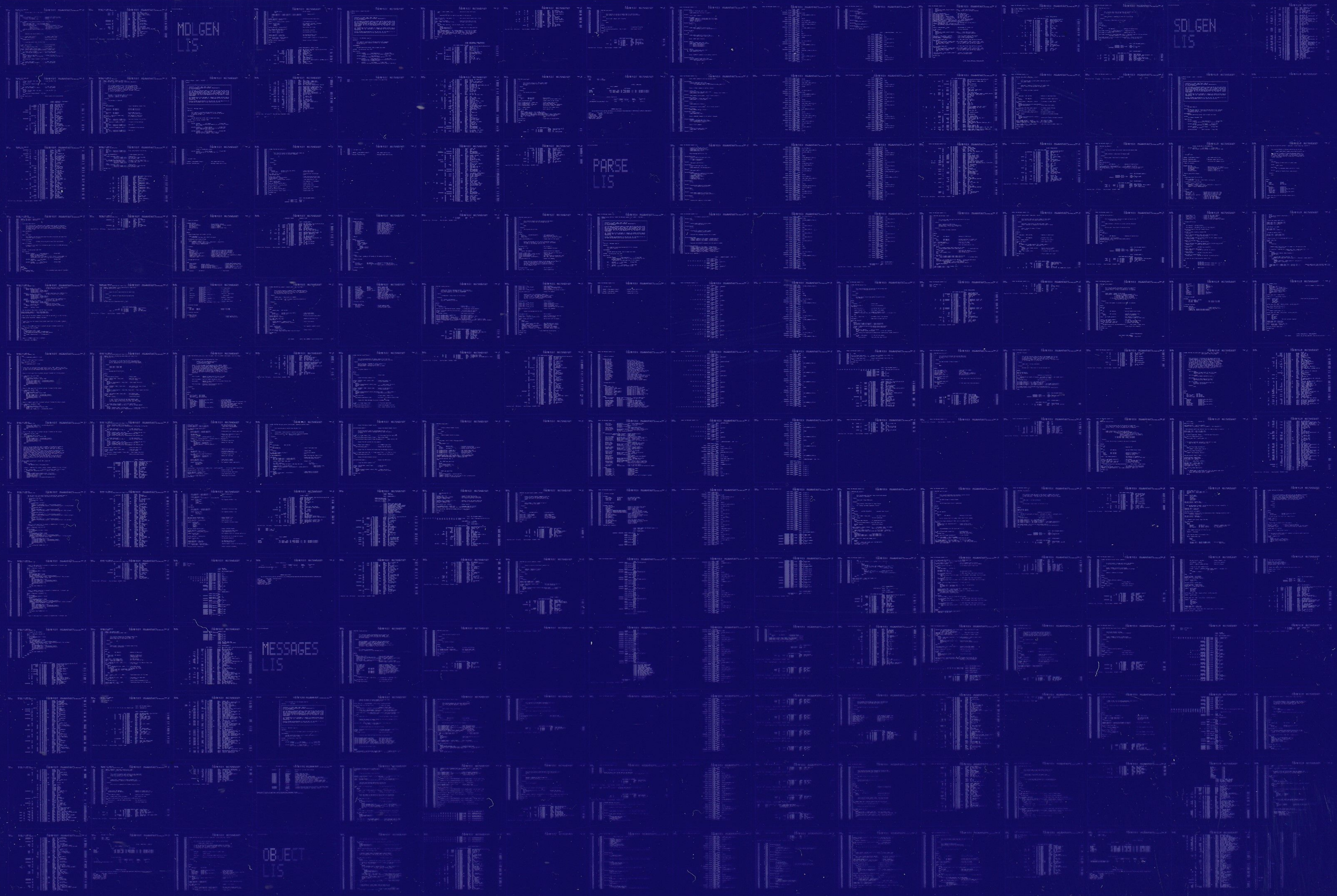
```
1 ROUTINE store_string =
1
1 !---
1 !
1 !       This routine stores the string value of a qualifier
1 !       into a specified location as long as the length is within
1 !       certain limits.  If not, a syntax error is signaled.
1 !
1 ! Inputs:
1 !
1 !       tpa$l_tokencnt/ptr = String to be stored
1 !       tpa$l_param = Address of a 3-longword argument list:
1 !                       1) Address of descriptor were the string
1 !                               length is stored in the first longword
1 !                               and the second longword is the place to
1 !                               store the string.
1 !                       2) Minimum legal length (unsigned)
1 !                       3) Maximum legal length (unsigned)
1 !
1 ! Outputs:
1 !
1 !       Either the string is stored or an error is signaled.
1 !---
1
2 BEGIN
2
2 BUILTIN
2     AP;                                 ! Register AP
2
2 MAP
2     AP:         REF BBLOCK;             ! Define TPARSE block format
2
2 LOCAL
2     args:       REF VECTOR,             ! Address of argument list
2     dest:       REF VECTOR,             ! Address of descriptor
2     length;                             ! Length of string
2
2 args = .ap [tpa$l_param]-4;             ! Get address of arguments
2                                         ! (PLIT value is addr of FIRST arg)
2
2 length = .ap [tpa$l_tokencnt];          ! Get length
2 IF .args [0] GEQ 2                      ! If 2nd, 3rd args specified,
2 THEN IF .length LSSU .args [2]          ! If less than minimum
2     OR .length GTRU .args [3]           ! or greater than maximum,
2 THEN
2     RETURN(syntax_error(.ap,emsg(symtoolng)));  ! then signal illegal value
2
2 dest = .args [1];                       ! Get address to store descriptor
2 dest [0] = .ap [tpa$l_tokencnt];        ! Store descriptor into quadword
2 CH$MOVE(.ap [tpa$l_tokencnt], .ap [tpa$l_tokenptr], .dest [1]);
2
2 RETURN true;
2
1 END;
```

```
                                                003C 00000 STORE_STRING:
                                                                   .WORD   Save R2,R3,R4,R5                          ; 1937
          52        20    AC              04   C3 00002            SUBL3   #4, 32(AP), ARGS                          ; 1974
                          50        10    AC   D0 00007            MOVL    16(AP), LENGTH                            ; 1977
                          02              62   D1 0000B            CMPL    (ARGS), #2                                ; 1978
                                          1A   19 0000E            BLSS    2$                                        ;
                    08    A2              50   D1 00010            CMPL    LENGTH, 8(ARGS)                           ; 1979
                                          06   1F 00014            BLSSU   1$                                        ;
                    0C    A2              50   D1 00016            CMPL    LENGTH, 12(ARGS)                          ; 1980
                                          0E   1B 0001A            BLEQU   2$                                        ;
                               00000000G  8F   DD 0001C 1$:        PUSHL   #MSG$_SYMTOOLNG                           ; 1982
                                          5C   DD 00022            PUSHL   AP                                        ;
                  0000G   CF              02   FB 00024            CALLS   #2, SYNTAX_ERROR                          ;
                                               04 00029            RET                                               ;
                          50        04    A2   D0 0002A 2$:        MOVL    4(ARGS), DEST                             ; 1984
                          60        10    AC   D0 0002E            MOVL    16(AP), (DEST)                            ; 1985
     04   B0        14    BC        10    AC   28 00032            MOVC3   16(AP), @20(AP), @4(DEST)                 ; 1986
                          50              01   D0 00039            MOVL    #1, R0                                    ; 1988
                                               04 0003C            RET                                               ; 1990

; Routine Size:  61 bytes,    Routine Base:  $CODE$ + 0911
; 1777        1991  1 GLOBAL ROUTINE allocate (bytes, address) =
; 1778        1992  1
; 1779        1993  1 !---
; 1780        1994  1 !
; 1781        1995  1 !       Allocate dynamic storage and return the address.
; 1782        1996  1 !       If an error occurs, the error is signaled.
; 1783        1997  1 !
; 1784        1998  1 ! Inputs:
; 1785        1999  1 !
; 1786        2000  1 !       bytes = Number of bytes to allocate
; 1787        2001  1 !       address = Longword to receive address of storage
; 1788        2002  1 !
; 1789        2003  1 ! Outputs:
; 1790        2004  1 !
; 1791        2005  1 !       address = Address of storage
; 1792        2006  1 !---
; 1793        2007  1
; 1794        2008  2 BEGIN
; 1795        2009  2
; 1796        2010  2 LOCAL
; 1797        2011  2     status;
; 1798        2012  2
; 1799        2013  2 status = lib$get_vm(bytes,.address);
; 1800        2014  2
; 1801        2015  2 IF NOT .status                  ! if unsuccessful,
; 1802        2016  2 THEN
; 1803        2017  2     SIGNAL(.status);            ! then signal the error
; 1804        2018  2
; 1805        2019  2 RETURN .status;                 ! return with status;
; 1806        2020  2
; 1807        2021  1 END;



                                    0004 00000          .ENTRY  ALLOCATE, Save R2                        ; 1991
                            08   AC  DD 00002           PUSHL   ADDRESS                                  ; 2013
                            04   AC  9F 00005           PUSHAB  BYTES                                    ;
            00000000G  00         02 FB 00008           CALLS   #2, LIB$GET_VM                           ;
                       52         50 D0 0000F           MOVL    R0, STATUS                               ;
                       09         52 E8 00012           BLBS    STATUS, 1$                               ; 2015
                                  52 DD 00015           PUSHL   STATUS                                   ; 2017
            00000000G  00         01 FB 00017           CALLS   #1, LIB$SIGNAL                           ;
                       50         52 D0 0001E 1$:       MOVL    STATUS, R0                               ; 2019
                                     04 00021           RET                                              ; 2021

; Routine Size:  34 bytes,    Routine Base:  $CODE$ + 094E
; 1809          2022  1 GLOBAL ROUTINE deallocate (bytes, address) =
; 1810          2023  1
; 1811          2024  1 !---
; 1812          2025  1 !
; 1813          2026  1 !       Deallocate dynamic storage.
; 1814          2027  1 !       If an error occurs, the error is signaled.
; 1815          2028  1 !
; 1816          2029  1 ! Inputs:
; 1817          2030  1 !
; 1818          2031  1 !       bytes = Number of bytes to deallocate
; 1819          2032  1 !       address = Address of storage to deallocate
; 1820          2033  1 !
; 1821          2034  1 ! Outputs:
; 1822          2035  1 !
; 1823          2036  1 !       None
; 1824          2037  1 !---
; 1825          2038  1
; 1826          2039  2 BEGIN
; 1827          2040  2
; 1828          2041  2 LOCAL
; 1829          2042  2     status;
; 1830          2043  2
; 1831          2044  2 status = lib$free_vm(bytes,address);
; 1832          2045  2
; 1833          2046  2 IF NOT .status                    ! if unsuccessful,
; 1834          2047  2 THEN
; 1835          2048  2     SIGNAL(.status);             ! then signal the error
; 1836          2049  2
; 1837          2050  2 RETURN .status;                   ! return with status;
; 1838          2051  2
; 1839          2052  1 END;



                                    0004 00000        .ENTRY  DEALLOCATE, Save R2                        ; 2022
                           08   AC   9F 00002        PUSHAB  ADDRESS                                    ; 2044
                           04   AC   9F 00005        PUSHAB  BYTES                                      ;
             00000000G  00          02   FB 00008        CALLS   #2, LIB$FREE_VM                            ;
                        52          50   D0 0000F        MOVL    R0, STATUS                                 ;
                        09          52   E8 00012        BLBS    STATUS, 1$                                 ; 2046
                                    52   DD 00015        PUSHL   STATUS                                     ; 2048
             00000000G  00          01   FB 00017        CALLS   #1, LIB$SIGNAL                             ;
                        50          52   D0 0001E 1$:    MOVL    STATUS, R0                                 ; 2050
                                    04 00021        RET                                                ; 2052

; Routine Size:  34 bytes,    Routine Base:  $CODE$ + 0970
; 1841          2053  1 GLOBAL ROUTINE comment =
; 1842          2054  1
; 1843          2055  1 !---
; 1844          2056  1 !
; 1845          2057  1 !       If MDL or SDL file is being generated, then output a
; 1846          2058  1 !       comment record for the appropriate file.
; 1847          2059  1 !
; 1848          2060  1 ! Inputs:
; 1849          2061  1 !
; 1850          2062  1 !       All inputs are drawn from tparse_block.
; 1851          2063  1 !
; 1852          2064  1 ! Outputs:
; 1853          2065  1 !
; 1854          2066  1 !       None
; 1855          2067  1 !
; 1856          2068  1 !---
; 1857          2069  1
; 1858          2070  2 BEGIN
; 1859          2071  2
; 1860          2072  2 LOCAL
; 1861          2073  2      comment_desc:    VECTOR[2];                      ! Comment descriptor
; 1862          2074  2
; 1863          2075  2 IF ( NOT .cli_flags [qual_mdl] ) AND                 ! If not generating MDL
; 1864          2076  3    ( NOT .cli_flags [qual_sdl] )                     !   and not gen'ing SDL
; 1865          2077  2 THEN                                                 !   then exit
; 1866          2078  2     RETURN true;
; 1867          2079  2
; 1868          2080  2 comment_desc[0] = .tparse_block[tpa$l_stringcnt];    ! Initialize comment_desc
; 1869          2081  2 comment_desc[1] = .tparse_block[tpa$l_stringptr];
; 1870          2082  2
; 1871          2083  2 IF .cli_flags [qual_mdl]
; 1872          2084  2 THEN
; 1873          2085  2     mdl_comment (comment_desc,                       ! Call mdl_comment with string
; 1874          2086  2             .new_line);
; 1875          2087  2
; 1876          2088  2 IF .cli_flags [qual_sdl]
; 1877          2089  2 THEN
; 1878          2090  2     sdl_comment (comment_desc,                       ! Call sdl_comment with string
; 1879          2091  2             .new_line);
; 1880          2092  2
; 1881          2093  2 new_line = true;                                     ! Reset new comment line indicator
; 1882          2094  2
; 1883          2095  2 RETURN true;
; 1884          2096  2
; 1885          2097  1 END;


                                  000C 00000          .ENTRY  COMMENT, Save R2,R3                          ; 2053
                 53     0000G  CF   9E 00002          MOVAB   CLI_FLAGS, R3                                ;
                 52     0000'  CF   9E 00007          MOVAB   NEW_LINE, R2                                 ;
                 5E            08   C2 0000C          SUBL2   #8, SP                                       ;
        04       63            03   E0 0000F          BBS     #3, CLI_FLAGS, 1$                            ; 2075
        24       63            04   E1 00013          BBC     #4, CLI_FLAGS, 4$                            ; 2076
                 6E     FDF8   C2   7D 00017 1$:      MOVQ    TPARSE_BLOCK+8, COMMENT_DESC                 ; 2080
```

```
                  0A            63              03  E1 0001C         BBC     #3, CLI_FLAGS, 2$                              ; 2083
                                                62  DD 00020         PUSHL   NEW_LINE                                       ; 2086
                                          04    AE  9F 00022         PUSHAB  COMMENT_DESC                                   ; 2085
                          0000G CF              02  FB 00025         CALLS   #2, MDL_COMMENT                                ;
                  0A            63              04  E1 0002A 2$:     BBC     #4, CLI_FLAGS, 3$                              ; 2088
                                                62  DD 0002E         PUSHL   NEW_LINE                                       ; 2091
                                          04    AE  9F 00030         PUSHAB  COMMENT_DESC                                   ; 2090
                          0000G CF              02  FB 00033         CALLS   #2, SDL_COMMENT                                ;
                                62              01  D0 00038 3$:     MOVL    #1, NEW_LINE                                   ; 2093
                                50              01  D0 0003B 4$:     MOVL    #1, R0                                         ; 2095
                                                    04 0003E         RET                                                    ; 2097

; Routine Size:  63 bytes,    Routine Base:  $CODE$ + 0992
; 1887          2098  1 END
; 1888          2099  0 ELUDOM



                                                                          .EXTRN  LIB$SIGNAL

;                                        PSECT SUMMARY
;
;       Name                        Bytes                      Attributes
;
;   $GLOBAL$                            240  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $OWN$                               992  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                             2513  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   _LIB$KEY0$                           54  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(1)
;   _LIB$STATE$                        1030  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(1)
;   $PLIT$                              348  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   _LIB$KEY1$                          216  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(1)




;                                 Library Statistics
;
;                                                   -------- Symbols --------        Pages        Processing
;       File                                        Total    Loaded   Percent        Mapped       Time
;
;   _$255$DUA28:[SYSLIB]STARLET.L32;1                9776        59         0          581          00:00.9
;   _$255$DUA28:[SYSLIB]TPAMAC.L32;1                   42        33        78           14          00:00.1




;                                        COMMAND QUALIFIERS

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

; Size:          2513 code + 2880 data bytes
; Run Time:         01:58.9
; Elapsed Time:     04:49.8
; Lines/CPU Min:     1058
; Lexemes/CPU-Min: 68377
; Memory Used:  450 pages
; Compilation Complete
```

0252 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



MDLGEN
LIS

SDLGEN
LIS

PARSE
LIS

MESSAGES
LIS

OBJECT
LIS