```
LLL              BBBBBBBBBBBB     RRRRRRRRRRRR
LLL              BBBBBBBBBBBB     RRRRRRRRRRRR
LLL              BBBBBBBBBBBB     RRRRRRRRRRRR
LLL              BBB         BBB  RR          RRR
LLL              BBB         BBB  RRR         RRR
LLL              BBB         BBB  RRR         RRR
LLL              BBB         BBB  RRR         RRR
LLL              BBB         BBB  RRR         RRR
LLL              BBB         BBB  RRR         RRR
LLL              BBBBBBBBBBBB     RRRRRRRRRRRR
LLL              BBBBBBBBBBBB     RRRRRRRRRRRR
LLL              BBBBBBBBBBBB     RRRRRRRRRRRR
LLL              BBB         BBB  RRR   RRR
LLL              BBB         BBB  RRR   RRR
LLL              BBB         BBB  RRR   RRR
LLL              BBB         BBB  RRR      RRR
LLL              BBB         BBB  RRR      RRR
LLL              BBB         BBB  RRR      RRR
LLLLLLLLLLLLLLL  BBBBBBBBBBBB     RRR         RRR
LLLLLLLLLLLLLLL  BBBBBBBBBBBB     RRR         RRR
LLLLLLLLLLLLLLL  BBBBBBBBBBBB     RRR         RRR
```

```
_$2

Val
---
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
002
00F
00F
00F
00F
00F
00F
00F
7FF
7FF
7FF
7FF
7FF
7FF
7FF
7FF
```

```
                                                                      J 12
**FILE**ID**OUTPUTHLP
```

```
  OOOOOO    UU      UU  TTTTTTTTTT  PPPPPPPP    UU      UU  TTTTTTTTTT  HH      HH  LL          PPPPPPPP
  OOOOOO    UU      UU  TTTTTTTTTT  PPPPPPPP    UU      UU  TTTTTTTTTT  HH      HH  LL          PPPPPPPP
OO      OO  UU      UU      TT      PP      PP  UU      UU      TT      HH      HH  LL          PP      PP
OO      OO  UU      UU      TT      PP      PP  UU      UU      TT      HH      HH  LL          PP      PP
OO      OO  UU      UU      TT      PP      PP  UU      UU      TT      HH      HH  LL          PP      PP
OO      OO  UU      UU      TT      PP      PP  UU      UU      TT      HH      HH  LL          PP      PP
OO      OO  UU      UU      TT      PPPPPPPP    UU      UU      TT      HHHHHHHHHH  LL          PPPPPPPP
OO      OO  UU      UU      TT      PPPPPPPP    UU      UU      TT      HHHHHHHHHH  LL          PPPPPPPP
OO      OO  UU      UU      TT      PP          UU      UU      TT      HH      HH  LL          PP
OO      OO  UU      UU      TT      PP          UU      UU      TT      HH      HH  LL          PP
OO      OO  UU      UU      TT      PP          UU      UU      TT      HH      HH  LL          PP              ....
OO      OO  UU      UU      TT      PP          UU      UU      TT      HH      HH  LL          PP              ....
  OOOOOO    UUUUUUUUUU      TT      PP          UUUUUUUUUU      TT      HH      HH  LLLLLLLLLL  PP              ....
  OOOOOO    UUUUUUUUUU      TT      PP          UUUUUUUUUU      TT      HH      HH  LLLLLLLLLL  PP              ....

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 lbr_outputhelp (
0                 LANGUAGE (BLISS32),
0                 IDENT = 'V04-000'
0                 ) =
1 BEGIN
1 %TITLE 'Prompting and library searching help function';
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 !
1 ! FACILITY:  Library access procedures.
1 !
1 ! ABSTRACT:
1 !
1 !       LBR$OUTPUT_HELP outputs help text to a user specified output
1 !       routine.  This information is drawn from an explicitly named
1 !       help library, or optionally, from user specified default help
1 !       libraries.  In addition, an optional prompting mode is available
1 !       so that LBR$OUTPUT_HELP can interact with a user and continue to
1 !       provide help information to a user after it has satisfied his
1 !       initial help request.
1 !
1 ! ENVIRONMENT:
1 !
1 !       VAX native, user mode.
1 !
1 !--
1 !
1 !
1 ! AUTHOR:  Peter George,        CREATION DATE:  01-May-1981
1 !
1 ! MODIFIED BY:
1 !
1 !       V03-005 GJA0081         Greg Awdziewicz         10-Apr-1984
1 !               - Reduce the severity of the signaled "openin" error
```

```
;   58          0058  1 !                       from fatal to error in the open_library routine.
;   59          0059  1 !                       - Return any error from open_library to the calling
;   60          0060  1 !                       program if the open fails for the specified main
;   61          0061  1 !                       library.
;   62          0062  1 !                       - Update the "ROUTINE VALUE:" comments to include the
;   63          0063  1 !                       codes "openin", "nohlplibs", and "usrinperr".
;   64          0064  1 !
;   65          0065  1 !               V03-004 GJA0077         Greg Awdziewicz         19-Feb-1984
;   66          0066  1 !                       - Set prompt_flags[hcf$v_more] when switching libraries
;   67          0067  1 !                       at the end of the search_libs routine to fix AccVio
;   68          0068  1 !                       which was occurring when the subprompt for a topic which
;   69          0069  1 !                       is in more than one library comes from the subtopic in
;   70          0070  1 !                       other than the main library when this subtopic was
;   71          0071  1 !                       requested at the subtopic level from the main library.
;   72          0072  1 !                       - Flag the end-topic case to further gate the output
;   73          0073  1 !                       of the additional libraries info.
;   74          0074  1 !                       - Allow filenames to be 39 characters long.
;   75          0075  1 !                       - Change the routine Make_upper_case to use a translate
;   76          0076  1 !                       table instead of conditionals inside a loop.
;   77          0077  1 !
;   78          0078  1 !               V03-003 JWT0066         Jim Teague              22-Nov-1982
;   79          0079  1 !                       Let RMS parse filenames.
;   80          0080  1 !
;   81          0081  1 !               V03-002 PCG0014         Peter George            06-Oct-1982
;   82          0082  1 !                       Modify library name substring search so that it only
;   83          0083  1 !                       checks for substrings that start from the beginning of
;   84          0084  1 !                       the file name.
;   85          0085  1 !
;   86          0086  1 !               V03-001 PCG0013         Peter George            01-Jul-1982
;   87          0087  1 !                       Make default search libraries work again by correcting
;   88          0088  1 !                       typographic error.
;   89          0089  1 !
;   90          0090  1 !--
;   92          0091   1 LIBRARY 'SYS$LIBRARY:STARLET';
;   93          0092   1 REQUIRE 'PREFIX';
;   94          0231   1 REQUIRE 'LBRDEF';
;   95          0822   1
;   96          0823   1 EXTERNAL ROUTINE
;   97          0824   1     lbr$close :         ADDRESSING_MODE(GENERAL),        ! Close library
;   98          0825   1     lbr$ini_control :   ADDRESSING_MODE(GENERAL),        ! Initialize librarian
;   99          0826   1     lbr$get_help :      ADDRESSING_MODE(GENERAL),        ! Get help text
;  100          0827   1     lbr$open :          ADDRESSING_MODE(GENERAL);        ! Open library
;  101          0828   1
;  102          0829   1 FORWARD ROUTINE
;  103          0830   1     prompt_help,                  ! Main prompting loop
;  104          0831   1     search_libs,                  ! Search default libraries till help is found
;  105          0832   1     change_lib,                   ! Change current help lib context to specified library
;  106          0833   1     switch_libname,               ! Update the library name descriptor
;  107          0834   1     tran_next_lib,                ! Get next user specified default help library
;  108          0835   1     open_library,                 ! Open a help library
;  109          0836   1     close_library,                ! Close a help library
;  110          0837   1     setup_keys,                   ! Set up help keys for lbr$get_help
;  111          0838   1     call_lbrhelp,                 ! Call lbr$get_help to get help from a library
;  112          0839   1     output_driver,                ! Driver for the user supplied output routine
;  113          0840   1     libs_available,               ! Output list of default libraries available
;  114          0841   1     file_present,                 ! Determine if file exists
;  115          0842   1     nohelp_log,                   ! Log unrecognized help requests
;  116          0843   1     remove_last_key,              ! Remove the last help key in a list of keys
;  117          0844   1     remove_terminator,            ! Remove the termination character at the end of a command
;  118          0845   1     make_upper_case;              ! Make a string all upper case
;  119          0846   1
;  120          0847   1 EXTERNAL LITERAL
;  121          0848   1     lbr$_endtopic,
;  122          0849   1     lbr$_illoutrou,
;  123          0850   1     lbr$_illinrou,
;  124          0851   1     lbr$_nohlplibs,
;  125          0852   1     lbr$_toomnyarg,
;  126          0853   1     lbr$_usrinperr;
;  127          0854   1
;  128          0855   1 LITERAL
;  129          0856   1     filename_length= 39,                                 ! Max filename length.
;  130          0857   1     main_libnumber = -2,                                 ! Library number of /LIBRARY specified library
;  131          0858   1     external_libnumber = -1;                             ! Library number of non-default, non-/LIB library
;  132          0859   1
;  133          0860   1 OWN
;  134          0861   1     end_topic_flag: BYTE,                                ! Flag aborts text.
;  135          0862   1     syshelp : COUNTEDSTRING ('SYS$HELP:.HLB'),           ! Default library spec
;  136          0863   1     topic : COUNTEDSTRING ('Topic? '),                   ! Topic prompt string
;  137          0864   1     subtopic : COUNTEDSTRING ('Subtopic? '),             ! Subtopic prompt string
;  138          0865   1     prompt_prefix : VECTOR [5, BYTE]                     ! Prompt control characters
;  139          0866   1                     INITIAL (BYTE (4, 13, 10, 13, 10)),
;  140          0867   1     control_flags : BBLOCK [hcf$c_length];               ! Global control flags
;  141          0868   1
;  142          0869   1 BIND
;  143          0870   1     help_flags = control_flags [hcf$l_userlib],          ! User default library flags
;  144          0871   1     prompt_flags = control_flags [hcf$l_prompt]: BLOCK[, BYTE]; ! Prompt control flags
;  145          0872   1
;  146          0873   1 MAP
;  147          0874   1     help_flags : BBLOCK;
```

```
1 %SBTTL 'Routine lbr$output_help';
1 GLOBAL ROUTINE lbr$output_help (output_rout, output_size, keys_desc,
1                                  library_desc, flags, input_rout) =
2 BEGIN
2
2 !++
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       This routine is the entry point for this module.  It performs
2 !       some initial processing on the input parameters, opens the main
2 !       help library, and then calls prompt_help to do all the real work.
2 !
2 ! CALLING SEQUENCE:
2 !
2 !       status = LBR$OUTPUT_HELP (output_routine, output_width [, [keys_desc]
2 !                                [, [library_desc]  [, [flags] [, [input_routine] ]]]] )
2 !
2 ! INPUTS:
2 !
2 !       output_rout =   address of user output routine
2 !       output_size =   width of the output line to be passed to the user
2 !                       output routine
2 !       keys_desc =     address of string desc for keys
2 !       library_desc =  address of string desc for help library name
2 !       flags =         address of longword of option flags
2 !       input_rout =    address of user input routine
2 !
2 ! OUTPUTS:
2 !
2 !       The requested help text is passed to the output routine.
2 !
2 ! ROUTINE VALUE:
2 !
2 !       status  lbr$_normal
2 !               lbr$_illoutrou   Output routine improperly specified or missing
2 !               lbr$_illinrou    Input routine improperly specified or missing
2 !               lbr$_toomnyarg   Too many arguments
2 !               lbr$_nohlplibs   No help libraries can be opened.
2 !               lbr$_usrinperr   Input error from user's action routine.
2 !
2 !       Some errors which occur may be signaled.  For example, if an error
2 !       occurs while trying to open the specified main default library
2 !       the error will be signaled as an "openin" error.  In this case
2 !       the return code value will have the inhibit bit set.  So the
2 !       return value will be
2 !
2 !       shr$_openin OR hlp$c_facility OR sts$k_error OR sts$m_inhib_msg
2 !--
2
2 MAP
2     keys_desc : REF BBLOCK,
2     library_desc : REF BBLOCK;
2
2 BUILTIN
2     ACTUALCOUNT,
2     NULLPARAMETER;
2
```

```
2 LOCAL
2       getcmd_line : BBLOCK [hlp$c_pagesize],                  ! Command line buffer
2       getcmd_desc : BBLOCK [dsc$c_s_bln],                     ! Command desc
2       indices : BBLOCK [hli$c_length],                        ! Help library indices
2       input_routine,                                          ! Address of user input routine
2       librarystring : BBLOCK [nam$c_maxrss],                  ! Default library name string
2       libraryname : BBLOCK [dsc$c_s_bln],                     ! String descriptor for library name
2       nomsg,                                                  ! Open_library message flag
2       output_routine,                                         ! Address of user output routine
2       output_width,                                           ! Width of output line
2       status;
2
2 BIND
2       main_libindex = indices [hli$l_mainindex],              ! Index of /LIB library
2       last_libindex = indices [hli$l_lastindex],              ! Index of last library examined
2       last_libnumber = indices [hli$l_lastnumb];              ! No. of last lib examined, relative to all default
2
2 help_flags = 0;                                               ! Clear help control flags
2
2 IF ACTUALCOUNT() GTR hlp$c_params                             ! If too many arguments
2    THEN RETURN lbr$_toomnyarg;                                !    then return error
2
2 IF NULLPARAMETER (hlp$c_outrout)                              ! If output_routine missing
2    OR .output_rout EQL 0                                      !    or zero
2    THEN RETURN lbr$_illoutrou                                 !    then return error
2    ELSE output_routine = .output_rout;                        !    else store data
2
2 IF NULLPARAMETER (hlp$c_outwidth)                             ! If output_width missing
2    OR ..output_size EQL 0                                     !    or zero
2    THEN output_width = hlp$c_liswidth                         !    then use default
2    ELSE output_width = ..output_size;                         !    else store data
2
2 getcmd_desc = 0;
2 getcmd_desc [dsc$a_pointer] = getcmd_line;
2 IF NOT NULLPARAMETER (hlp$c_liredesc)                         ! If keys_desc present
3    THEN BEGIN                                                 !    then pick up passed descriptor
3         getcmd_desc [dsc$w_length] = .keys_desc [dsc$w_length];
3         CH$MOVE (.getcmd_desc [dsc$w_length],                 !    fill buffer
3                  .keys_desc [dsc$a_pointer],
3                  .getcmd_desc [dsc$a_pointer] );
2         END;
2
2 libraryname = 0;
2 libraryname [dsc$a_pointer] = librarystring;                  ! Init pointer to library name buffer
2 IF NOT NULLPARAMETER (hlp$c_libname)                          ! If library_name specified
3    THEN BEGIN                                                 !    then override default
3         help_flags [hlp$v_library] = true;
3         libraryname [dsc$w_length] = .library_desc [dsc$w_length];
3         CH$MOVE (.libraryname [dsc$w_length],
3                  .library_desc [dsc$a_pointer],
3                  .libraryname [dsc$a_pointer]);
3         END
2    ELSE help_flags [hlp$v_library] = false;
2
2 IF NOT NULLPARAMETER (hlp$c_flags)                            ! If flags present
2    THEN help_flags = .help_flags                              !    then get relevent bits
3                  OR (hlp$m_prompt AND ..flags)
```

```
3                     OR (hlp$m_process AND ..flags)
3                     OR (hlp$m_group AND ..flags)
3                     OR (hlp$m_system AND ..flags)
3                     OR (hlp$m_liblist AND ..flags)
3                     OR (hlp$m_help AND ..flags)
2      ELSE help_flags = .help_flags OR hlp$m_prompt        !    else set defaults
2                     OR hlp$m_process OR hlp$m_group
2                     OR hlp$m_system AND NOT hlp$m_liblist
2                     AND NOT hlp$m_help;
2
2  IF .help_flags [hlp$v_prompt]                            ! If prompting enabled
3  THEN BEGIN
4        IF ( NULLPARAMETER (hlp$c_inrout)                  ! And output_routine missing
4            OR .input_rout EQL 0 )                         ! or zero
3          THEN RETURN lbr$_illinrou                        !    then return error
3          ELSE input_routine = .input_rout;                !    else store data
3        prompt_flags = hcf$m_cont;                         ! Turn on prompting
3        END
2  ELSE prompt_flags = hcf$m_noprompt;                      ! else turn off prompting
2
2  IF .help_flags [hlp$v_process] OR                        ! If default lib searching enabled
2     .help_flags [hlp$v_group] OR .help_flags [hlp$v_system]
2     THEN help_flags = .help_flags OR hlp$m_all            !    then set all flag
2     ELSE help_flags = .help_flags AND NOT hlp$m_all;      !    else clear all flag
2
2  IF .help_flags [hlp$v_library]                           ! If library specified
3     THEN BEGIN                                            ! Then open it
3         LOCAL
3             local_status;
3         nomsg = false;                                    ! Signal error if library can't be opened
3         local_status = open_library (main_libindex, libraryname, .nomsg);  ! Open main library
3         IF NOT .local_status THEN RETURN .local_status;  ! Return any errors.
3         last_libindex = .main_libindex;                   ! Set last library used to main library
3         last_libnumber = main_libnumber;
3         END
3
3     ELSE BEGIN                                            ! Else use a default library
3
3         LOCAL
3             libno,
3             acmode;
3
3         IF NOT .help_flags [hlp$v_all]                    ! Are we allowed to?
3           THEN RETURN lbr$_nohlplibs;                     ! If not then signal error
3         libno = -1;                                       ! Initialize search
3         status = false;                                   ! Init while condition
3         nomsg = true;                                     ! Do not signal open errors
3         WHILE NOT .status                                 ! While more libraries
4         DO BEGIN                                          ! Get and try to open one
4            IF NOT tran_next_lib (libraryname, acmode, libno)
4               THEN RETURN lbr$_nohlplibs;
4            status = open_library (last_libindex, libraryname, .nomsg);
3            END;
3         last_libnumber = 0;                               ! Set lib number
2         END;
2
2 !
```

```
:  320          1046  2 ! Call prompt_help to do the real help work.
:  321          1047  2 !
:  322          1048  2 status = prompt_help(getcmd_desc, output_width, .input_routine, .output_routine,
:  323          1049  2                      indices, libraryname);
:  324          1050  2
:  325          1051  2 IF .help_flags [hlp$v_library]                          ! If library specified
:  326          1052  2    THEN close_library (main_libindex);                  ! Close the main library
:  327          1053  2
:  328          1054  2 RETURN .status
:  329          1055  2
:  330          1056  1 END;                                                    ! Of lbr$output_help


                                                          .TITLE  LBR_OUTPUTHELP Prompting and library searching
;                                                                                help function
                                                          .IDENT  \V04-000\

                                                          .PSECT  $OWN$,NOEXE,2

                                          00000 END_TOPIC_FLAG:
                                                          .BLKB   1
                                          00001           .BLKB   3
                                      0D  00004 SYSHELP:.BYTE   13                                      ;
   42  4C  48  2E  3A  50  4C  45  48  24  53  59  53  00005  .ASCII  \SYS$HELP:.HLB\                          ;
                                          00012           .BLKB   2
                                      07  00014 TOPIC:  .BYTE   7                                       ;
                    20  3F  63  69  70  6F  54  00015  .ASCII  \Topic? \                                ;
                                      0A  0001C SUBTOPIC:
                                                          .BYTE   10                                      ;
         20  3F  63  69  70  6F  74  62  75  53  0001D  .ASCII  \Subtopic? \                             ;
                                          00027           .BLKB   1
                         0A  0D  0A  0D  04  00028 PROMPT_PREFIX:
                                                          .BYTE   4, 13, 10, 13, 10                       ;
                                          0002D           .BLKB   3
                                          00030 CONTROL_FLAGS:
                                                          .BLKB   8

                                                HELP_FLAGS=         CONTROL_FLAGS
                                                PROMPT_FLAGS=       CONTROL_FLAGS+4
                                                          .EXTRN  LBR$CLOSE, LBR$INI_CONTROL
                                                          .EXTRN  LBR$GET_HELP, LBR$OPEN
                                                          .EXTRN  LBR$_ENDTOPIC, LBR$_ILLOUTROU
                                                          .EXTRN  LBR$_ILLINROU, LBR$_NOHLPLIBS
                                                          .EXTRN  LBR$_TOOMNYARG, LBR$_USRINPERR

                                                          .PSECT  $CODE$,NOWRT,2

                                 00FC  00000           .ENTRY  LBR$OUTPUT_HELP, Save R2,R3,R4,R5,R6,R7     ; 0876
               57       0000'  CF  9E  00002           MOVAB   HELP_FLAGS, R7                              ;
               5E       FCD8   CE  9E  00007           MOVAB   -808(SP), SP                                ;
                               67  D4  0000C           CLRL    HELP_FLAGS                                  ; 0949
               06              6C  91  0000E           CMPB    (AP), #6                                    ; 0951
                               08  1B  00011           BLEQU   1$                                          ;
               50 00000000G    8F  D0  00013           MOVL    #LBR$_TOOMNYARG, R0                         ; 0952
                                   04  0001A           RET                                                 ;
                               6C  95  0001B 1$:       TSTB    (AP)                                        ; 0954
                               05  13  0001D           BEQL    2$                                          ;
```

```
                                 04  AC  D5 0001F          TSTL    4(AP)                                          ;
                                 08  12 00022              BNEQ    3$                                             ;
                    50 00000000G 8F  D0 00024 2$:          MOVL    #LBR$_ILLOUTROU, R0                            ; 0956
                                     04 0002B              RET                                                    ;
                    56       04  AC  D0 0002C 3$:          MOVL    OUTPUT_ROUT, OUTPUT_ROUTINE                    ; 0957
                    02           6C  91 00030              CMPB    (AP), #2                                       ; 0959
                                 0A  1F 00033              BLSSU   4$                                             ;
                             08  AC  D5 00035              TSTL    8(AP)                                          ;
                                 05  13 00038              BEQL    4$                                             ;
                             08  BC  D5 0003A              TSTL    @OUTPUT_SIZE                                   ; 0960
                                 07  12 0003D              BNEQ    5$                                             ;
               08   AE       50  8F  9A 0003F 4$:          MOVZBL  #80, OUTPUT_WIDTH                              ; 0961
                                 05  11 00044              BRB     6$                                             ;
               08   AE       08  BC  D0 00046 5$:          MOVL    @OUTPUT_SIZE, OUTPUT_WIDTH                     ; 0962
                           0120  CE  D4 0004B 6$:          CLRL    GETCMD_DESC                                    ; 0964
             0124   CE     0128  CE  9E 0004F              MOVAB   GETCMD_LINE, GETCMD_DESC+4                     ; 0965
                    03           6C  91 00056              CMPB    (AP), #3                                       ; 0966
                                 17  1F 00059              BLSSU   7$                                             ;
                             0C  AC  D5 0005B              TSTL    12(AP)                                         ;
                                 12  13 0005E              BEQL    7$                                             ;
                    50       0C  AC  D0 00060              MOVL    KEYS_DESC, R0                                  ; 0968
             0120   CE           60  B0 00064              MOVW    (R0), GETCMD_DESC                              ;
      0124  DE      04  B0     0120  CE  28 00069          MOVC3   GETCMD_DESC, @4(R0), @GETCMD_DESC+4            ; 0971
                             0C  AE  D4 00072 7$:          CLRL    LIBRARYNAME                                    ; 0974
               10   AE       14  AE  9E 00075              MOVAB   LIBRARYSTRING, LIBRARYNAME+4                   ; 0975
                    04           6C  91 0007A              CMPB    (AP), #4                                       ; 0976
                                 1A  1F 0007D              BLSSU   8$                                             ;
                             10  AC  D5 0007F              TSTL    16(AP)                                         ;
                                 15  13 00082              BEQL    8$                                             ;
               01   A7           04  88 00084              BISB2   #4, HELP_FLAGS+1                               ; 0978
                    50       10  AC  D0 00088              MOVL    LIBRARY_DESC, R0                               ; 0979
               0C   AE           60  B0 0008C              MOVW    (R0), LIBRARYNAME                              ;
        10  BE      04  B0       0C  AE  28 00090          MOVC3   LIBRARYNAME, @4(R0), @LIBRARYNAME+4            ; 0982
                                 04  11 00097              BRB     9$                                             ; 0976
               01   A7           04  8A 00099 8$:          BICB2   #4, HELP_FLAGS+1                               ; 0984
                    05           6C  91 0009D 9$:          CMPB    (AP), #5                                       ; 0986
                                 4D  1F 000A0              BLSSU   10$                                            ;
                             14  AC  D5 000A2              TSTL    20(AP)                                         ;
                                 48  13 000A5              BEQL    10$                                            ;
   50      14  BC            01      00  EF 000A7          EXTZV   #0, #1, @FLAGS, R0                             ; 0988
                    50           67  C8 000AD              BISL2   HELP_FLAGS, R0                                 ;
          51   14   BC FFFFFFFD  8F  CB 000B0              BICL3   #-3, @FLAGS, R1                                ; 0989
                    51           50  C8 000B9              BISL2   R0, R1                                         ;
          50   14   BC FFFFFFFB  8F  CB 000BC              BICL3   #-5, @FLAGS, R0                                ; 0990
                    50           51  C8 000C5              BISL2   R1, R0                                         ;
          51   14   BC FFFFFFF7  8F  CB 000C8              BICL3   #-9, @FLAGS, R1                                ; 0991
                    51           50  C8 000D1              BISL2   R0, R1                                         ;
          50   14   BC FFFFFFEF  8F  CB 000D4              BICL3   #-17, @FLAGS, R0                               ; 0992
                    50           51  C8 000DD              BISL2   R1, R0                                         ;
          51   14   BC FFFFFFDF  8F  CB 000E0              BICL3   #-33, @FLAGS, R1                               ; 0993
          67        51           50  C9 000E9              BISL3   R0, R1, HELP_FLAGS                             ;
                                 03  11 000ED              BRB     11$                                            ; 0987
                    67           0F  88 000EF 10$:         BISB2   #15, HELP_FLAGS                                ; 0996
                    1C           67  E9 000F2 11$:         BLBC    HELP_FLAGS, 14$                                ; 0999
                    06           6C  91 000F5              CMPB    (AP), #6                                       ; 1001
                                 05  1F 000F8              BLSSU   12$                                            ;
                             18  AC  D5 000FA              TSTL    24(AP)                                         ;
```

```
                                  08  12 000FD         BNEQ    13$                                      ;
                     50 00000000G 8F  D0 000FF 12$:    MOVL    #LBR$_ILLINROU, R0                       ; 1003
                                      04 00106         RET
                     54       18  AC  D0 00107 13$:    MOVL    INPUT_ROUT, INPUT_ROUTINE                ; 1004
              04     A7           01  D0 0010B         MOVL    #1, PROMPT_FLAGS                         ; 1005
                                  04  11 0010F         BRB     15$                                      ; 0999
              04     A7           01  CE 00111 14$:    MNEGL   #1, PROMPT_FLAGS                         ; 1007
       08            67           01  E0 00115 15$:    BBS     #1, HELP_FLAGS, 16$                      ; 1009
       04            67           02  E0 00119         BBS     #2, HELP_FLAGS, 16$                      ; 1010
       06            67           03  E1 0011D         BBC     #3, HELP_FLAGS, 17$                      ;
              01     A7           08  88 00121 16$:    BISB2   #8, HELP_FLAGS+1                         ; 1011
                                  04  11 00125         BRB     18$                                      ;
              01     A7           08  8A 00127 17$:    BICB2   #8, HELP_FLAGS+1                         ; 1012
       21     01     A7           02  E1 0012B 18$:    BBC     #2, HELP_FLAGS+1, 19$                    ; 1014
                                  52  D4 00130         CLRL    NOMSG                                    ; 1018
                                  52  DD 00132         PUSHL   NOMSG                                    ; 1019
                          10      AE  9F 00134         PUSHAB  LIBRARYNAME                              ;
                          011C    CE  9F 00137         PUSHAB  MAIN_LIBINDEX                            ;
            0000V    CF           03  FB 0013B         CALLS   #3, OPEN_LIBRARY                         ;
                     76           50  E9 00140         BLBC    LOCAL_STATUS, 26$                        ; 1020
            0118     CE   0114    CE  D0 00143         MOVL    MAIN_LIBINDEX, LAST_LIBINDEX             ; 1021
            011C     CE           02  CE 0014A         MNEGL   #2, LAST_LIBNUMBER                       ; 1022
                                  3D  11 0014F         BRB     24$                                      ; 1014
       19     01     A7           03  E1 00151 19$:    BBC     #3, HELP_FLAGS+1, 21$                    ; 1031
                     6E           01  CE 00156         MNEGL   #1, LIBNO                                ; 1033
                     52           01  7D 00159         MOVQ    #1, NOMSG                                ; 1035
                     2B           53  E8 0015C 20$:    BLBS    STATUS, 23$                              ; 1036
                                  5E  DD 0015F         PUSHL   SP                                       ; 1038
                          08      AE  9F 00161         PUSHAB  ACMODE                                   ;
                          14      AE  9F 00164         PUSHAB  LIBRARYNAME                              ;
            0000V    CF           03  FB 00167         CALLS   #3, TRAN_NEXT_LIB                        ;
                     08           50  E8 0016C         BLBS    R0, 22$                                  ;
                     50 00000000G 8F  D0 0016F 21$:    MOVL    #LBR$_NOHLPLIBS, R0                      ; 1039
                                      04 00176         RET
                                  52  DD 00177 22$:    PUSHL   NOMSG                                    ; 1040
                          10      AE  9F 00179         PUSHAB  LIBRARYNAME                              ;
                          0120    CE  9F 0017C         PUSHAB  LAST_LIBINDEX                            ;
            0000V    CF           03  FB 00180         CALLS   #3, OPEN_LIBRARY                         ;
                     53           50  D0 00185         MOVL    R0, STATUS                               ;
                                  D2  11 00188         BRB     20$                                      ; 1036
                          011C    CE  D4 0018A 23$:    CLRL    LAST_LIBNUMBER                           ; 1042
                          0C      AE  9F 0018E 24$:    PUSHAB  LIBRARYNAME                              ; 1048
                          0118    CE  9F 00191         PUSHAB  INDICES                                  ;
                          0050    8F  BB 00195         PUSHR   #^M<R4,R6>                               ;
                          18      AE  9F 00199         PUSHAB  OUTPUT_WIDTH                             ;
                          0134    CE  9F 0019C         PUSHAB  GETCMD_DESC                              ;
            0000V    CF           06  FB 001A0         CALLS   #6, PROMPT_HELP                          ;
                     53           50  D0 001A5         MOVL    R0, STATUS                               ;
       09     01     A7           02  E1 001A8         BBC     #2, HELP_FLAGS+1, 25$                    ; 1051
                          0114    CE  9F 001AD         PUSHAB  MAIN_LIBINDEX                            ; 1052
            0000V    CF           01  FB 001B1         CALLS   #1, CLOSE_LIBRARY                        ;
                     50           53  D0 001B6 25$:    MOVL    STATUS, R0                               ; 1054
                                      04 001B9 26$:    RET                                              ; 1056

; Routine Size:  442 bytes,    Routine Base:  $CODE$ + 0000
```

```
1 %SBTTL 'Routine prompt_help';
1 ROUTINE prompt_help (getcmd_desc, output_width, input_routine,
1                      output_routine, indices, libraryname) =
2 BEGIN
2
2 !++
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       This routine contains the interactive code loop that is repeatedly
2 !       executed when help is being run in prompting mode.  This same loop
2 !       is executed exactly once when noprompt is specified.
2 !
2 ! INPUTS:
2 !
2 !       getcmd_desc =   address of the descriptor for the set of keys
2 !                       to be processed
2 !
2 !       output_width =  address of longword containing width of output line
2 !
2 !       input_routine = address of user supplied input routine
2 !
2 !       output_routine = addresss of user supplied output routine
2 !
2 !       indices =       address of data structure containing indices of
2 !                       libraries currently in use
2 !
2 !       libraryname =   address of string desc for default help library name
2 !
2 ! OUTPUTS:
2 !
2 !       None.
2 !
2 ! ROUTINE VALUE:
2 !
2 !       Always true.
2 !
2 !--
2 MAP
2     getcmd_desc : REF BBLOCK,
2     indices : REF BBLOCK,
2     libraryname : REF BBLOCK;
2
2 LOCAL
2     char_pos,                                          ! Position of first non-blank character in input str
2     key_descs : VECTOR [dsc$c_s_bln * hlp$c_maxkeys, BYTE],     !String descriptors for keys
2     key_length_array : VECTOR [hlp$c_maxkeys],         ! Array of key lengths
2     lib_name : BBLOCK [dsc$c_s_bln],                   ! Library name descriptor
2     lib_name_buf : VECTOR [filename_length, BYTE],     ! Library name buffer
2     print_data : BBLOCK [hpd$c_length],                ! Data structure for output driver
2     topic_prompt : BBLOCK [dsc$c_s_bln],               ! Topic prompt descriptor
2     topic_prompt_buf : BBLOCK [hlp$c_pagesize],        ! Topic prompt buffer
2     sub_prompt : BBLOCK [dsc$c_s_bln],                 ! Sub-prompt descriptor
2     sub_prompt_level,                                  ! Current key depth
2     sub_prompt_line : BBLOCK [hlp$c_pagesize],         ! Sub-prompt line
2     sub_prompt_buf : BBLOCK [hlp$c_pagesize],          ! Sub-prompt buffer
2     status;
2
```

```
BIND
    main_libindex = indices [hli$l_mainindex],          ! Index of /LIB library
    last_libindex = indices [hli$l_lastindex],          ! Index of last library examined
    last_libnumber = indices [hli$l_lastnumb],          ! No. of last lib examined, relative to all default
    getcmd_line = .getcmd_desc [dsc$a_pointer],         ! Command buffer
    true_keys = print_data [hpd$b_truekeys] : SIGNED BYTE,  ! Number of help keys
    help_level = print_data [hpd$b_helplevel] : BYTE,   ! Current key depth
    print_flags = print_data [hpd$b_printflag] : BBLOCK,! Flags for output driver
    add_info_level = print_data [hpd$l_subpmtlev],      ! Level of additional info
    sub_prompt_ptr = print_data [hpd$l_subpmtptr],      ! Ptr used for filling sub-prompt buffer
    length_array = print_data [hpd$l_lenarray] : REF VECTOR,  ! Address of key length array
    outputroutine = print_data [hpd$l_outputrou];       ! User specified output routine

length_array = key_length_array;                                            ! Init print_data array
outputroutine = .output_routine;

lib_name = 0;                                                               ! Initialize library name
lib_name [dsc$a_pointer] = lib_name_buf;
IF NOT .help_flags [hlp$v_library]                                          ! If no main library
   THEN switch_libname (.libraryname, lib_name);                            ! Then use passed library name in pr

topic_prompt = 0;                                                           ! Initialize topic prompt
topic_prompt [dsc$a_pointer] = topic_prompt_buf;
CH$MOVE (.prompt_prefix [0], prompt_prefix [1], topic_prompt_buf);

sub_prompt = 0;                                                             ! Initialize sub-prompt
sub_prompt [dsc$a_pointer] = sub_prompt_line;
CH$MOVE (.prompt_prefix [0], prompt_prefix [1], sub_prompt_line);

WHILE (.getcmd_desc [dsc$w_length] GTR 0)                                   ! Remove any preceeding blanks from
      AND (CH$RCHAR (.getcmd_desc [dsc$a_pointer]) EQL %C' ')
DO BEGIN
   getcmd_desc [dsc$w_length] = .getcmd_desc [dsc$w_length] - 1;
   getcmd_desc [dsc$a_pointer] = .getcmd_desc [dsc$a_pointer] + 1;
   END;

IF (.getcmd_desc [dsc$w_length] GTR 0)                                      ! If non-empty help keys
   AND (CH$RCHAR (.getcmd_desc [dsc$a_pointer]) EQL %C'@')                  ! And initial key starts with "@"
   THEN (change_lib (.getcmd_desc, .getcmd_desc [dsc$a_pointer],
                     .indices, lib_name));                                  ! Then change to specified library

WHILE (.prompt_flags AND hcf$m_cont) NEQ 0                                  ! WHILE we can still continue...
DO BEGIN

IF .prompt_flags EQL hcf$m_noprompt                                         ! If one shot command
   THEN prompt_flags = NOT hcf$m_noprompt;                                  ! Execute while exactly once

IF .prompt_flags GTR hcf$m_stay                                             ! If prompting enabled
   THEN BEGIN                                                               ! Prompt for keywords

!
! If moving down a prompt level then update the subtopic prompt.
!

IF (.prompt_flags AND (hcf$m_more OR hcf$m_stay)) EQL hcf$m_more
   THEN BEGIN

```

```
5          LOCAL
5              sub_length;
5
5          IF .lib_name [dsc$w_length] NEQ 0                                  ! If not main library
6             THEN BEGIN                                                      ! Then insert library name in prompt
6                  CH$WCHAR (%C'@', sub_prompt_line + 4);
6                  CH$MOVE (.lib_name [dsc$w_length],
6                           .lib_name [dsc$a_pointer],
6                           sub_prompt_line + 4 + 1);
6                  CH$WCHAR (%X'20', sub_prompt_line + 4 + 1 + .lib_rame [dsc$w_length]);
6                  sub_prompt [dsc$w_length] = 4 + 1 + .lib_name [dsc$w_length] + 1;
6                  END
5             ELSE sub_prompt [dsc$w_length] = 4;                             ! Otherwise skip to keys
5
5          sub_length = .sub_prompt_ptr - sub_prompt_buf;                     ! Calculate length of keys
5          CH$MOVE (.sub_length, sub_prompt_buf,                              ! Move keys into prompt line
5                   sub_prompt_line + .sub_prompt [dsc$w_length]);
5          CH$MOVE (.subtopic [0], subtopic [1],                              !Move "subtopic?" into prompt line
5                   sub_prompt_line + .sub_prompt [dsc$w_length] + .sub_length);
5          sub_prompt [dsc$w_length] = .sub_prompt [dsc$w_length]             ! Update prompt length
5                                      + .sub_length + .subtopic [0];
5          sub_prompt_level = .help_level;                                    ! Update prompt level
4          END;
4
4  !
4  ! If subprompt length greater than zero, then do ~ubtopic prompting.
4  !
4
4  IF .sub_prompt [dsc$w_length] NEQ 0
5     THEN BEGIN
5
5          LOCAL
5              prefix_length,
5              sub_length;
5
5          IF .lib_name [dsc$w_length] NEQ 0                                  ! If not main library
5             THEN prefix_length = 4 + 1 + .lib_name [dsc$w_length] + 1       ! Then prefix includes lib name
5             ELSE prefix_length = 4;                                         ! Else it doesn't
5
5          sub_length = .sub_prompt [dsc$w_length] - .prefix_length           ! Calculate length of keys
5                       - .subtopic [0];
5          CH$MOVE (.sub_length, sub_prompt_line + .prefix_length,            ! Move keys into command line
5                       getcmd_line);
5          getcmd_desc [dsc$w_length] = hlp$c_pagesize - .sub_length;         ! Calculate space left in command li
5          getcmd_desc [dsc$a_pointer] = getcmd_line + .sub_length;           ! Set pointer to end of keys
5
6          IF (status = (.input_routine) (getcmd_desc [dsc$w_length],         ! Get input
5               sub_prompt, getcmd_desc [dsc$w_length])) EQL RMS$_EOF         ! If CNTL-Z
5             THEN EXITLOOP                                                   ! Then get out of help
5             ELSE IF NOT .status                                             ! Else if other error
5                     THEN RETURN lbr$_usrinperr;                             ! Then signal user input error
5
7          IF ((char_pos = CH$FIND_NOT_CH (.getcmd_desc [dsc$w_length],       ! If blank line
6                 .getcmd_desc [dsc$a_pointer], %ASCII ' ')) EQL 0)           ! Or CR
6             OR (CH$RCHAR (.char_pos) EQL %X'0D')                            ! Then back up a prompt level
6
5             THEN IF (sub_prompt_level = .sub_prompt_level - 1) EQL 0        ! If now at topic level
```

```
5                     THEN sub_prompt [dsc$w_length] = 0                     ! then clear subtopic prompt
6                     ELSE BEGIN
6                          remove_last_key (sub_prompt,                      ! else remove last key
6                                  .key_length_array [.sub_prompt_level+1]);
6                          prompt_flags = hcf$m_backup OR hcf$m_cont;        ! Set backup flag
6                          END
6
6              ELSE SELECTONE (CH$RCHAR (.getcmd_desc [dsc$w_length] +       ! Test termination character
5                          .getcmd_desc [dsc$a_pointer] - 1)) OF SET
5
5                  [%C'?']:                                                  ! If ?
6                  BEGIN                                                     ! then repeat help for old help keys
6                  getcmd_desc [dsc$w_length] = .sub_length;
6                  getcmd_desc [dsc$a_pointer] = getcmd_line;
5                  END;
5
5                  [%X'1A']:                                                 ! If CNTL-Z
5                  EXITLOOP;                                                 ! then get out of help
5
5                  [OTHERWISE]:                                              ! If help keys
6                  BEGIN                                                     ! then append command to old keys
6                  getcmd_desc [dsc$w_length] = .getcmd_desc [dsc$w_length] + .sub_length;
6                  getcmd_desc [dsc$a_pointer] = getcmd_line;
5                  END;
5                  TES;
5
5           remove_terminator (.getcmd_desc);                                ! Remove termination character from
5
4           END;
4
4 !
4 ! If subtopic length equals zero, then do topic prompting
4 !
4
4 IF .sub_prompt [dsc$w_length] EQL 0
5    THEN BEGIN
5
5           getcmd_desc [dsc$w_length] = hlp$c_pagesize;                     ! Init rest of descriptor
5           getcmd_desc [dsc$a_pointer] = getcmd_line;
5
5           IF .last_libnumber GEQ external_libnumber                        ! If not main library
6              THEN BEGIN                                                    ! Then insert library name in prompt
6                  CH$WCHAR (%C'@', topic_prompt_buf + 4);
6                  CH$MOVE (.lib_name [dsc$w_length],
6                          .lib_name [dsc$a_pointer],
6                          topic_prompt_buf + 4 + 1);
6                  CH$WCHAR (%X'20', topic_prompt_buf + 4 + 1 +
6                          .lib_name [dsc$w_length]);
6                  CH$MOVE (.topic [0], topic [1],
6                          topic_prompt_buf + 4 + 1 + .lib_name [dsc$w_length] + 1);
6                  topic_prompt [dsc$w_length] = 4 + 1 +
6                          .lib_name [dsc$w_length] + 1 + .topic [0];
6                  END
6              ELSE BEGIN                                                    ! Otherwise, do not include library
6                  topic_prompt [dsc$w_length] = 4 + .topic [0];
6                  CH$MOVE (.topic [0], topic [1],
6                          topic_prompt_buf + 4);
```

```
5                   END;
5
6               IF (status = (.input_routine) (getcmd_desc [dsc$w_length],      ! Get input
5                   topic_prompt, getcmd_desc [dsc$w_length])) EQL RMS$_EOF     ! If CNTL-Z
5                  THEN EXITLOOP                                                ! Then get out of help
5                  ELSE IF NOT .status                                          ! Else if other error
5                       THEN RETURN lbr$_usrinperr;                              ! Then signal user input error
5
7               IF ((char_pos = CH$FIND_NOT_CH (.getcmd_desc [dsc$w_length],    ! If blank line
6                       .getcmd_desc [dsc$a_pointer], %ASCII ' ')) EQL 0)        ! Or CR
6                  OR (CH$RCHAR (.char_pos) EQL %X'0D')                          ! Then back up a library
7                  THEN (IF (.last_libnumber LSS external_libnumber              ! If already at main library
7                           OR NOT .help_flags [hlp$v_library])                  ! or if there is no main library
6                       THEN EXITLOOP                                            ! Then get out of help
7                       ELSE BEGIN                                               ! Else back up to main library
7                           close_library (last_libindex);                       ! Close old user library
7                           last_libnumber = main_libnumber;                     ! Set main library number
7                           last_libindex = .main_libindex;                      ! And index
7                           lib_name [dsc$w_length] = 0;                         ! Reset library name
7                           getcmd_desc [dsc$w_length] = 0;                      ! Clear command descriptor
7                           getcmd_desc [dsc$a_pointer] = getcmd_line;
7                           prompt_flags = hcf$m_backup OR hcf$m_cont;           ! Set back up flag
6                           END;)
6
6                  ELSE SELECTONE (CH$RCHAR (.getcmd_desc [dsc$w_length] +       ! Test termination character
5                           .getcmd_desc [dsc$a_pointer] - 1)) OF SET
5
5                   [%C'?']:                                                     ! If ?
6                   BEGIN                                                        ! Then repeat help for old library
6                   getcmd_desc [dsc$w_length] = 0;
6                   getcmd_desc [dsc$a_pointer] = getcmd_line;
5                   END;
5
5                   [%X'1A']:                                                    ! If CNTL-Z
5                   EXITLOOP;                                                    ! Then get out of help
5
5                   [OTHERWISE]:                                                 ! If help keys
6                   IF (CH$RCHAR (.char_pos) EQL %C'@')                          ! And start with "@"
6                      THEN (change_lib (.getcmd_desc, .char_pos,
5                                        .indices, lib_name));                   ! Then change to specified library
5                   TES;
5
5               remove_terminator (.getcmd_desc);                                ! Remove termination character from
5
4               END;
4
3       END;                                                                     ! Of prompt for keywords
3
3       IF (.prompt_flags AND hcf$m_backup) EQL 0                                ! If not backing up a level
4       THEN BEGIN                                                               ! then get help text
5           IF .prompt_flags NEQ (NOT hcf$m_noprompt)                            ! If not prompting
4              THEN prompt_flags = hcf$m_cont;                                   ! Say to continue prompting
4           sub_prompt_ptr = sub_prompt_buf;                                     ! Init subtopic buffer pointer
4           help_level = 0;                                                      ! Set current key depth to zero
4           add_info_level = 0;
4           setup_keys (.getcmd_desc, key_descs, true_keys);                     ! Set up individual help key descs
4           print_flags [hpd$v_init] = 0;                                        ! Set print not initialized flag
```

```
:  617        1342  4        end_topic_flag = false;                                          ! Clear flag which aborts text.
:  618        1343  5        IF NOT (status = search_libs (key_descs, .output_width, print_data,! Search the libraries for help
:  619        1344  5                .indices, lib_name, .sub_prompt [dsc$w_length]))
:  620        1345  4           THEN RETURN .status;
:  621        1346  4        IF .print_flags [hpd$v_init]                                     ! If no help found
:  622        1347  4           THEN nohelp_log (.getcmd_desc);                               ! Then log help request
:  623        1348  4        IF .help_flags [hlp$v_liblist] AND                               ! Output list of libraries if condit
:  624        1349  4           NOT .end_tcpic_flag AND                                       ! are right
:  625        1350  5           ((.true_keys LEQ 0) OR
:  626        1351  5            ((.add_info_level EQL 0) AND .print_flags [hpd$v_init]))
:  627        1352  4        THEN
:  628        1353  4           libs_available (.output_routine, .output_width);
:  629        1354  4        END
:  630        1355  3 ELSE prompt_flags = hcf$m_cont OR hcf$m_stay;                           ! Say to continue prompting at curre
:  631        1356  3
:  632        1357  2 END;                                                                    !Of while
:  633        1358  2
:  634        1359  2 RETURN true
:  635        1360  2
:  636        1361  1 END;                                                                    !Of prompt_help



                                                OFFC 00000 PROMPT_HELP:
                                                                    .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11            ; 1058
                              5E     F92C   CE   9E 00002           MOVAB    -1748(SP), SP                                   ;
                              5A       14   AC   D0 00007           MOVL     INDICES, R10                                    ; 1115
                              58       04   AC   D0 0000B           MOVL     GETCMD_DESC, R8                                 ; 1118
                              59       04   A8   9E 0000F           MOVAB    4(R8), R9                                       ;
                                            69   DD 00013           PUSHL    (R9)                                            ;
                        FF50   CD      88   AD   9E 00015           MOVAB    KEY_LENGTH_ARRAY, LENGTH_ARRAY                  ; 1127
                        FF4C   CD      10   AC   D0 0001B           MOVL     OUTPUT_ROUTINE, OUTPUTROUTINE                   ; 1128
                                       80   AD   D4 00021           CLRL     LIB_NAME                                        ; 1130
                          84   AD    FF58   CD   9E 00024           MOVAB    LIB_NAME_BUF, LIB_NAME+4                        ; 1131
                 0B    0000'   CF            02  E0 0002A           BBS      #2, HELP_FLAGS+1, 1$                            ; 1132
                                       80   AD   9F 00030           PUSHAB   LIB_NAME                                        ; 1133
                                       18   AC   DD 00033           PUSHL    LIBRARYNAME                                     ;
                        0000V  CF           02   FB 00036           CALLS    #2, SWITCH_LIBNAME                              ;
                                     FF3C   CD   D4 0003B 1$:       CLRL     TOPIC_PROMPT                                    ; 1135
                        FF40   CD    FD3C   CD   9E 0003F           MOVAB    TOPIC_PROMPT_BUF, TOPIC_PROMPT+4                ; 1136
                               50    0000'  CF   9A 00046           MOVZBL   PROMPT_PREFIX, R0                               ; 1137
           FD3C    CD   0000'  CF           50   28 0004B           MOVC3    R0, PROMPT_PREFIX+1, TOPIC_PROMPT_BUF           ;
                                     FD34   CD   D4 00053           CLRL     SUB_PROMPT                                      ; 1139
                        FD38   CD    020C   CE   9E 00057           MOVAB    SUB_PROMPT_LINE, SUB_PROMPT+4                   ; 1140
                               50    0000'  CF   9A 0005E           MOVZBL   PROMPT_PREFIX, R0                               ; 1141
           020C    CE   0000'  CF           50   28 00063           MOVC3    R0, PROMPT_PREFIX+1, SUB_PROMPT_LINE            ;
                                            68   B5 0006B 2$:       TSTW     (R8)                                            ; 1143
                                            0C   13 0006D           BEQL     3$                                              ;
                               20      00   B9   91 0006F           CMPB     @0(R9), #32                                     ; 1144
                                            06   12 00073           BNEQ     3$                                              ;
                                            68   B7 00075           DECW     (R8)                                            ; 1146
                                            69   D6 00077           INCL     (R9)                                            ; 1147
                                            F0   11 00079           BRB      2$                                              ; 1143
                                            68   B5 0007B 3$:       TSTW     (R8)                                            ; 1150
                                            15   13 0007D           BEQL     4$                                              ;
                        40    8F          00   B9   91 0007F          CMPB    @0(R9), #64                                  ; 1151
                                               0E   12 00084          BNEQ    4$                                           ;
                                          80   AD   9F 00086          PUSHAB  LIB_NAME                                     ; 1152
                                               5A   DD 00089          PUSHL   R10                                          ; 1153
                                               69   DD 0008B          PUSHL   (R9)                                         ; 1152
                                               58   DD 0008D          PUSHL   R8                                           ;
                     0000V    CF               04   FB 0008F          CALLS   #4, CHANGE_LIB                               ;
                              03       0000'   CF   E8 00094 4$:      BLBS    PROMPT_FLAGS, 5$                             ; 1155
                                             02D4   31 00099          BRW     45$                                          ;
                  FFFFFFFF    8F       0000'   CF   D1 0009C 5$:      CMPL    PROMPT_FLAGS, #-1                            ; 1158
                                               04   12 000A5          BNEQ    6$                                           ;
                                       0000'   CF   D4 000A7          CLRL    PROMPT_FLAGS                                 ; 1159
                              02       0000'   CF   D1 000AB 6$:      CMPL    PROMPT_FLAGS, #2                             ; 1161
                                               03   14 000B0          BGTR    7$                                           ;
                                             021F   31 000B2          BRW     37$                                          ;
               50      0000'  CF FFFFFFF9      8F   CB 000B5 7$:      BICL3   #-7, PROMPT_FLAGS, R0                        ; 1168
                              04               50   D1 000BF          CMPL    R0, #4                                       ;
                                               5E   12 000C2          BNEQ    10$                                          ;
                              56          80   AD   3C 000C4          MOVZWL  LIB_NAME, R6                                 ; 1174
                                               1B   13 000C8          BEQL    8$                                           ;
                      0210    CE          40   8F   90 000CA          MOVB    #64, SUB_PROMPT_LINE+4                       ; 1176
        0211    CE      84    BD               56   28 000D0          MOVC3   R6, @LIB_NAME+4, SUB_PROMPT_LINE+5           ; 1179
                      0211  CE46               20   90 000D7          MOVB    #32, SUB_PROMPT_LINE+5[R6]                   ; 1180
        FD34    CD            56               06   A1 000DD          ADDW3   #6, R6, SUB_PROMPT                           ; 1181
                                               05   11 000E3          BRB     9$                                           ; 1174
                      FD34    CD               04   B0 000E5 8$:      MOVW    #4, SUB_PROMPT                               ; 1183
                              50          0C   AE   9E 000EA 9$:      MOVAB   SUB_PROMPT_BUF, R0                           ; 1185
               57     FF44    CD               50   C3 000EE          SUBL3   R0, SUB_PROMPT_PTR, SUB_LENGTH               ;
                              56        FD34   CD   3C 000F4          MOVZWL  SUB_PROMPT, R6                               ; 1187
        020C  CE46      0C    AE               57   28 000F9          MOVC3   SUB_LENGTH, SUB_PROMPT_BUF, SUB_PROMPT_LINE- ;
                                                                              [R6]
                              50       0000'   CF   9A 00101          MOVZBL  SUBTOPIC, R0                                 ; 1188
                              56               57   C0 00106          ADDL2   SUB_LENGTH, R6                               ; 1189
        020C  CE46     0000'  CF               50   28 00109          MOVC3   R0, SUBTOPIC+1, SUB_PROMPT_LINE[R6]          ;
                              50       0000'   CF   9A 00112          MOVZBL  SUBTOPIC, R0                                 ; 1191
        FD34    CD            56               50   A1 00117          ADDW3   R0, R6, SUB_PROMPT                           ;
                              5B        FF55   CD   9A 0011D          MOVZBL  HELP_LEVEL, SUB_PROMPT_LEVEL                 ; 1192
                              51        FD34   CD   3C 00122 10$:     MOVZWL  SUB_PROMPT, R1                               ; 1199
                                               03   12 00127          BNEQ    11$                                          ;
                                             00A9   31 00129          BRW     23$                                          ;
                                          80   AD   B5 0012C 11$:     TSTW    LIB_NAME                                     ; 1206
                                               09   13 0012F          BEQL    12$                                          ;
                              50          80   AD   3C 00131          MOVZWL  LIB_NAME, PREFIX_LENGTH                      ; 1207
                              50               06   C0 00135          ADDL2   #6, PREFIX_LENGTH                            ;
                                               03   11 00138          BRB     13$                                          ;
                              50               04   D0 0013A 12$:     MOVL    #4, PREFIX_LENGTH                            ; 1208
                              51               50   C2 0013D 13$:     SUBL2   PREFIX_LENGTH, R1                            ; 1210
                              56       0000'   CF   9A 00140          MOVZBL  SUBTOPIC, SUB_LENGTH                         ; 1211
               56             51               56   C3 00145          SUBL3   SUB_LENGTH, R1, SUB_LENGTH                   ;
          00    BE     020C  CE40              56   28 00149          MOVC3   SUB_LENGTH, SUB_PROMPT_LINE-                 ; 1212
                                                                              [PREFIX_LENGTH], @0(SP)
               68     0200    8F               56   A3 00151          SUBW3   SUB_LENGTH, #512, (R8)                       ; 1214
               69             6E               56   C1 00157          ADDL3   SUB_LENGTH, (SP), (R9)                       ; 1215
                                               58   DD 0015B          PUSHL   R8                                           ; 1218
                                        FD34   CD   9F 0015D          PUSHAB  SUB_PROMPT                                   ; 1217
                                               58   DD 00161          PUSHL   R8                                           ; 1218
                      0C      BC               03   FB 00163          CALLS   #3, @INPUT_ROUTINE                           ;
```

```
                04   AE          50  D0 00167          MOVL    R0, STATUS                              :
          0001827A   8F     04   AE  D1 0016B          CMPL    STATUS, #98938                          :
                                 51  13 00173          BEQL    20$                                     :
                     03     04   AE  E8 00175          BLBS    STATUS, 14$                             : 1220
                               00D1  31 00179          BRW     28$                                     :
    00   B9          68          20  3B 0017C  14$:    SKPC    #32, (R8), @0(R9)                       : 1223
                                 02  12 00181          BNEQ    15$                                     :
                                 51  D4 00183          CLRL    R1                                      :
                08   AE          51  D0 00185  15$:    MOVL    R1, CHAR_POS                            :
                                 06  13 00189          BEQL    16$                                     : 1224
                     0D     08   BE  91 0018B          CMPB    @CHAR_POS, #13                          : 1225
                                 1E  12 0018F          BNEQ    18$                                     :
                                 5B  D7 00191  16$:    DECL    SUB_PROMPT_LEVEL                        : 1227
                                 06  12 00193          BNEQ    17$                                     :
                          FD34   CD  B4 00195          CLRW    SUB_PROMPT                              : 1228
                                 33  11 00199          BRB     22$                                     :
                             8C AD4B  DD 0019B  17$:    PUSHL   KEY_LENGTH_ARRAY+4[SUB_PROMPT_LEVEL]    : 1231
                          FD34   CD  9F 0019F          PUSHAB  SUB_PROMPT                              : 1230
             0000V   CF          02  FB 001A3          CALLS   #2, REMOVE_LAST_KEY                     :
             0000'   CF          11  D0 001A8          MOVL    #17, PROMPT_FLAGS                       : 1232
                                 1F  11 001AD          BRB     22$                                     : 1227
                     50          68  3C 001AF  18$:    MOVZWL  (R8), R0                                : 1236
                     50          69  C0 001B2          ADDL2   (R9), R0                                :
                     50     FF   A0  9A 001B5          MOVZBL  -1(R0), R0                              :
                     3F          50  91 001B9          CMPB    R0, #63                                 : 1238
                                 05  12 001BC          BNEQ    19$                                     :
                     68          56  B0 001BE          MOVW    SUB_LENGTH, (R8)                        : 1240
                                 08  11 001C1          BRB     21$                                     : 1241
                     1A          50  91 001C3  19$:    CMPB    R0, #26                                 : 1244
                                 7F  13 001C6  20$:    BEQL    27$                                     :
                     68          56  A0 001C8          ADDW2   SUB_LENGTH, (R8)                        : 1249
                     69          6E  D0 001CB  21$:    MOVL    (SP), (R9)                              : 1250
                                 58  DD 001CE  22$:    PUSHL   R8                                      : 1254
             0000V   CF          01  FB 001D0          CALLS   #1, REMOVE_TERMINATOR                   :
                          FD34   CD  B5 001D5  23$:    TSTW    SUB_PROMPT                              : 1262
                                 03  13 001D9          BEQL    24$                                     :
                               00F6  31 001DB          BRW     37$                                     :
                     68   0200   8F  B0 001DE  24$·    MOVW    #512, (R8)                              : 1265
                     69          6E  D0 001E3          MOVL    (SP), (R9)                              : 1266
                     57   0000'  CF  9A 001E6          MOVZBL  TOPIC, R7                               : 1276
          FFFFFFFF   8F     08   AA  D1 001EB          CMPL    8(R10), #-1                             : 1268
                                 2C  19 001F3          BLSS    25$                                     :
              FD40   CD     40   8F  90 001F5          MOVB    #64, TOPIC_PROMPT_BUF+4                 : 1270
                     56     80   AD  3C 001FB          MOVZWL  LIB_NAME, R6                            : 1271
    FD41   CD        84     BD   56  28 001FF          MOVC3   R6, @LIB_NAME+4, TOPIC_PROMPT_BUF+5     : 1273
              FD41 CD46          20  90 00206          MOVB    #32, TOPIC_PROMPT_BUF+5[R6]             : 1275
    FD42 CD46        0000'  CF   57  28 0020C          MOVC3   R7, TOPIC+1, TOPIC_PROMPT_BUF+6[R6]     : 1277
                     50  06 A746  9E 00215          MOVAB   6(R7)[R6], R0                           : 1279
              FF3C   CD          50  B0 0021A          MOVW    R0, TOPIC_PROMPT                        :
                                 0E  11 0021F          BRB     26$                                     : 1268
    FF3C   CD        57          04  A1 00221  25$:    ADDW3   #4, R7, TOPIC_PROMPT                    : 1282
    FD40   CD     0000'  CF      57  28 00227          MOVC3   R7, TOPIC+1, TOPIC_PROMPT_BUF+4         : 1284
                                 58  DD 0022F  26$:    PUSHL   R8                                      : 1288
                          FF3C   CD  9F 00231          PUSHAB  TOPIC_PROMPT                            : 1287
                                 58  DD 00235          PUSHL   R8                                        1288
                0C   BC          03  FB 00237          CALLS   #3, @INPUT_ROUTINE                      :
                04   AE          50  D0 0023B          MOVL    R0, STATUS                              :
                 0001827A  8F        04  AE  D1 0023F          CMPL    STATUS, #98938                      :
                                         2B  13 00247 27$:     BEQL    32$                                 :
                           08        04  AE  E8 00249          BLBS    STATUS, 29$                         : 1290
                           50 00000000G  8F  D0 0024D 28$:     MOVL    #LBR$_USRINPERR, R0                 : 1291
                                             04 00254          RET                                         :
             00   B9       68            20  3B 00255 29$:     SKPC    #32, (R8), @0(R9)                   : 1293
                                         02  12 0025A          BNEQ    30$                                 :
                                         51  D4 0025C          CLRL    R1                                  :
                       08  AE            51  D0 0025E 30$:     MOVL    R1, CHAR_POS                        :
                                         06  13 00262          BEQL    31$                                 : 1294
                           0D        08  BE  91 00264          CMPB    @CHAR_POS, #13                      : 1295
                                         32  12 00268          BNEQ    34$                                 :
                 FFFFFFFF  8F        08  AA  D1 0026A 31$:     CMPL    8(R10), #-1                         : 1296
                                         03  18 00272          BGEQ    33$                                 :
                                       00F9  31 00274 32$:     BRW     45$                                 :
                  F7     0000'  CF       02  E1 00277 33$:     BBC     #2, HELP_FLAGS+1, 32$               : 1297
                                     04  AA  9F 0027D          PUSHAB  4(R10)                              : 1300
                       0000V  CF         01  FB 00280          CALLS   #1, CLOSE_LIBRARY                   :
                          08  AA         02  CE 00285          MNEGL   #2, 8(R10)                          : 1301
                          04  AA         6A  D0 00289          MOVL    (R10), 4(R10)                       : 1302
                                     80  AD  B4 0028D          CLRW    LIB_NAME                            : 1303
                                         68  B4 00290          CLRW    (R8)                                : 1304
                           69            6E  D0 00292          MOVL    (SP), (R9)                          : 1305
                       0000'  CF         11  D0 00295          MOVL    #17, PROMPT_FLAGS                   : 1306
                                         31  11 0029A          BRB     36$                                 : 1293
                           50            68  3C 0029C 34$:     MOVZWL  (R8), R0                            : 1310
                           50            69  C0 0029F          ADDL2   (R9), R0                            :
                           50        FF  A0  9A 002A2          MOVZBL  -1(R0), R0                          :
                           3F            50  91 002A6          CMPB    R0, #63                             : 1312
                                         07  12 002A9          BNEQ    35$                                 :
                                         68  B4 002AB          CLRW    (R8)                                : 1314
                           69            6E  D0 002AD          MOVL    (SP), (R9)                          : 1315
                                         1B  11 002B0          BRB     36$                                 : 1309
                           1A            50  91 002B2 35$:     CMPB    R0, #26                             : 1318
                                         BD  13 002B5          BEQL    32$                                 :
                          40  8F     08  BE  91 002B7          CMPB    @CHAR_POS, #64                      : 1322
                                         0F  12 002BC          BNEQ    36$                                 :
                                     80  AD  9F 002BE          PUSHAB  LIB_NAME                            : 1323
                                         5A  DD 002C1          PUSHL   R10                                 : 1324
                                     10  AE  DD 002C3          PUSHL   CHAR_POS                            : 1323
                                         58  DD 002C6          PUSHL   R8                                  :
                       0000V  CF         04  FB 002C8          CALLS   #4, CHANGE_LIB                      :
                                         58  DD 002CD 36$:     PUSHL   R8                                  : 1327
                       0000V  CF         01  FB 002CF          CALLS   #1, REMOVE_TERMINATOR               :
                  03   0000'  CF         04  E1 002D4 37$:     BBC     #4, PROMPT_FLAGS, 38$               : 1333
                                       008B  31 002DA          BRW     43$                                 :
                                  0000'  CF  D5 002DD 38$:     TSTL    PROMPT_FLAGS                        : 1335
                                         05  13 002E1          BEQL    39$                                 :
                       0000'  CF         01  D0 002E3          MOVL    #1, PROMPT_FLAGS                    : 1336
                        FF44  CD     0C  AE  9E 002E8 39$:     MOVAB   SUB_PROMPT_BUF, SUB_PROMPT_PTR      : 1337
                                   FF55  CD  94 002EE          CLRB    HELP_LEVEL                          : 1338
                                   FF48  CD  D4 002F2          CLRL    ADD_INFO_LEVEL                      : 1339
                                   FF54  CD  9F 002F6          PUSHAB  TRUE_KEYS                           : 1340
                                     B0  AD  9F 002FA          PUSHAB  KEY_DESCS                           :
                                         58  DD 002FD          PUSHL   R8                                  :
                       0000V  CF         03  FB 002FF          CALLS   #3, SETUP_KEYS                      :
                        FF56  CD         01  8A 00304          BICB2   #1, PRINT_FLAGS                     : 1341
```

```
                              0000'  CF  94 00309          CLRB    END_TOPIC_FLAG                          ; 1342
                    7E        FD34   CD  3C 0030D          MOVZWL  SUB_PROMPT, -(SP)                       ; 1344
                                80   AD  9F 00312          PUSHAB  LIB_NAME                                ; 1343
                                5A   DD 00315              PUSHL   R10                                     ; 1344
                              FF44   CD  9F 00317          PUSHAB  PRINT_DATA                              ; 1343
                                08   AC  DD 0031B          PUSHL   OUTPUT_WIDTH                            ;
                                B0   AD  9F 0031E          PUSHAB  KEY_DESCS                               ;
              0000V  CF              06  FB 00321          CALLS   #6, SEARCH_LIBS                         ;
                 04  AE              50  D0 00326          MOVL    R0, STATUS                              ;
                     05         04   AE  E8 0032A          BLBS    STATUS, 40$                             ;
                     50         04   AE  D0 0032E          MOVL    STATUS, R0                              ; 1345
                                         04 00332          RET                                             ;
                     07       FF56   CD  E9 00333 40$:     BLBC    PRINT_FLAGS, 41$                        ; 1346
                                     58  DD 00338          PUSHL   R8                                      ; 1347
              0000V  CF              01  FB 0033A          CALLS   #1, NOHELP_LOG                          ;
        28    0000'  CF              04  E1 0033F 41$:     BBC     #4, HELP_FLAGS, 44$                     ; 1348
                     23       0000'  CF  E8 00345          BLBS    END_TOPIC_FLAG, 44$                     ; 1349
                              FF54   CD  95 0034A          TSTB    TRUE_KEYS                               ; 1350
                                     0B  15 0034E          BLEQ    42$                                     ;
                              FF48   CD  D5 00350          TSTL    ADD_INFO_LEVEL                          ; 1351
                                     17  12 00354          BNEQ    44$                                     ;
                     12       FF56   CD  E9 00356          BLBC    PRINT_FLAGS, 44$                        ;
                                08   AC  DD 0035B 42$:     PUSHL   OUTPUT_WIDTH                            ; 1353
                                10   AC  DD 0035E          PUSHL   OUTPUT_ROUTINE                          ;
              0000V  CF              02  FB 00361          CALLS   #2, LIBS_AVAILABLE                      ;
                                     05  11 00366          BRB     44$                                     ; 1333
              0000'  CF              03  D0 00368 43$:     MOVL    #3, PROMPT_FLAGS                        ; 1355
                                   FD24  31 0036D 44$:     BRW     4$                                      ; 1155
                     50              01  D0 00370 45$:     MOVL    #1, R0                                  ; 1359
                                         04 00373          RET                                             ; 1361

; Routine Size:  884 bytes,    Routine Base:  $CODE$ + 01BA
```

```
1 %SBTTL 'Routine search_libs';
1 ROUTINE search_libs (keydescs, outputwidth, printdata,
1                 indices, libname, subpromptlen) =
2 BEGIN
2
2 !++
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       This routine searches the default help libraries and calls the
2 !       librarian function to extract help from each help library as
2 !       required.
2 !
2 ! INPUTS:
2 !
2 !       keydescs =      address of vector of key descriptors
2 !
2 !       outputwidth =   address of longword containing width of output line
2 !
2 !       printdata =     address of data structure containing info for
2 !                       the output driver
2 !
2 !       indices =       address of data structure containing indices of
2 !                       libraries currently in use
2 !
2 !       libname =       address of descriptor for the user default library name
2 !
2 !       subpromptlen =  total length of help keys, zero => at topic prompt level
2 !
2 ! OUTPUTS:
2 !
2 !       printdata =     flags are manipulated by this routine and other
2 !                       values are altered by the output driver
2 !
2 !       indices =       updated to reflect library that help information
2 !                       was eventually extracted from
2 !
2 !       libname =       if help found in default user library, updated to
2 !                       the file name of that library
2 !
2 ! ROUTINE VALUE:
2 !
2 !       Always true.
2 !
2 !--
2 MAP
2     indices : REF BBLOCK,
2     keydescs : REF BBLOCK,
2     libname : REF BBLOCK,
2     printdata : REF BBLOCK;
2
2 BIND
2     main_libindex = indices [hli$l_mainindex],          ! Index of /LIB library
2     last_libindex = indices [hli$l_lastindex],          ! Index of last library examined
2     last_libnumber = indices [hli$l_lastnumb],          ! No. of last library examined, relative to all defa
2     print_flags = printdata [hpd$b_printflag] : BBLOCK; ! Flags for output driver
2
2 LOCAL
```

```
2      current_libindex,                                    ! Index of library currently being searched
2      current_libnumber,                                   ! Number of library currently being searched
2      deflib_acmode : BYTE,                                ! Logical name table number
2      librarystring : BBLOCK [nam$c_maxrss],               ! Default library name string
2      libraryname : BBLOCK [dsc$c_s_bln],                  ! String descriptor for library name
2      nomsg,                                               ! Open library message flag
2      user_libno,                                          ! HLP$LIBRARY number
2      status;
2
2  nomsg = true;                                            ! Do not signal error if library can't be opened
2  print_flags [hpd$v_found] = false;                       ! Initialize help found flag
2  libraryname [dsc$a_pointer] = librarystring;             ! Initialize descriptor
2
3  IF (NOT .subpromptlen)                                   ! If at TOPIC level
3     AND ((.help_flags AND hlp$m_all) NEQ 0)               ! and default searching enabled
3     AND (.keydescs [dsc$w_length] NEQ 0)                  ! and keys are non-empty
2     THEN print_flags [hpd$v_all] = false                  ! then print help only if found
2     ELSE print_flags [hpd$v_all] = true;                  ! else print help always
2
3  IF NOT (status = call_lbrhelp (last_libindex,
3                  .outputwidth, .printdata, .keydescs))
2     THEN RETURN (.status);
2
2  IF .print_flags [hpd$v_found]                            ! If help was found
3     THEN BEGIN                                            ! then done
3          print_flags [hpd$v_found] = false;               ! Reset flag
3          RETURN true;                                     ! Exit
2          END;
2
2  print_flags [hpd$v_all] = false;                         ! Print help if found
2
2  IF .help_flags [hlp$v_library]                           ! Check /LIB library
3     AND (.last_libnumber GEQ 0)                           ! if it exists and
3     THEN BEGIN                                            ! if haven't already
3          current_libindex = .main_libindex;               ! Init libindex
4          IF NOT (status = call_lbrhelp (last_libindex,
4                  .outputwidth, .printdata, .keydescs))
3             THEN RETURN (.status);
2          END;
2
2  IF .print_flags [hpd$v_found]                            ! If help found
2     THEN current_libnumber = main_libnumber               ! Then set libnumber to main library
2     ELSE current_libnumber = -1;                          ! Else prepare to search default libraries
2
2  user_libno = -1;                                         ! Init library translation
2  WHILE (NOT .print_flags [hpd$v_found]) AND               ! Search libraries
3        (tran_next_lib (libraryname, deflib_acmode, user_libno) NEQ 0)
3  DO BEGIN
3     current_libnumber = .current_libnumber + 1;           ! Increment current libnumber
3     IF .current_libnumber NEQ .last_libnumber             ! If lib not already open and searched
4        THEN IF (open_library (current_libindex, libraryname, .nomsg))    ! If library successfully opened
4                THEN BEGIN
4                     status = call_lbrhelp (current_libindex,
4                             .outputwidth, .printdata, .keydescs);
4                     IF NOT .print_flags [hpd$v_found]                ! If help not found
4                        THEN close_library (current_libindex);        ! Then close library
4                     IF NOT .status THEN RETURN (.status);            ! If error then return
```

```
;   752      1476 3                      END;
;   753      1477 2          END;
;   754      1478 2
;   755      1479 2 IF NOT .print_flags [hpd$v_found]                     ! If help still not found
;   756      1480 3     THEN BEGIN                                        ! Then go back to last library
;   757      1481 3          current_libindex = .last_libindex;           ! Set libindex
;   758      1482 3          current_libnumber = .last_libnumber;         ! Set libnumber
;   759      1483 3          print_flags [hpd$v_all] = true;              ! Print help not available message
;   760      1484 4          IF NOT (status = call_lbrhelp (last_libindex,
;   761      1485 4                  .outputwidth, .printdata, .keydescs))
;   762      1486 3              THEN RETURN (.status);
;   763      1487 2          END;
;   764      1488 2
;   765      1489 3 IF (.current_libnumber NEQ .last_libnumber)           ! If help found in new library
;   766      1490 3     THEN BEGIN                                        ! Then clean up
;   767      1491 4          IF (.last_libnumber GTR external_libnumber)  ! If last library not main library
;   768      1492 3              THEN close_library (last_libindex);      ! Then close that library
;   769      1493 4          IF (.current_libnumber GTR external_libnumber)  ! If new library not main library
;   770      1494 3          THEN
;   771      1495 4               BEGIN
;   772      1496 4               switch_libname (libraryname, .libname);  ! Then change library name and
;   773      1497 4               prompt_flags[hcf$v_more] = true;         ! indicate that there's more.
;   774      1498 4               END
;   775      1499 3          ELSE                                         ! Otherwise, simply reset it
;   776      1500 3               libname [dsc$w_length] = 0;
;   777      1501 3          last_libindex = .current_libindex;           ! Reset last libindex
;   778      1502 3          last_libnumber = .current_libnumber;         ! Reset last libnumber
;   779      1503 2          END;
;   780      1504 2
;   781      1505 2 RETURN true;
;   782      1506 1 END;                                                  ! Of search_libs



                                                  03FC 00000 SEARCH_LIBS:
                                                                     .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9       ; 1363
                      59      0000V   CF   9E 00002                  MOVAB    CALL_LBRHELP, R9                   ;
                      5E      FEEC    CE   9E 00007                  MOVAB    -276(SP), SP                       ;
       57      10     AC              04   C1 0000C                  ADDL3    #4, INDICES, R7                    ; 1414
       56      10     AC              08   C1 00011                  ADDL3    #8, INDICES, R6                    ; 1415
                      53       0C     AC   D0 00016                  MOVL     PRINTDATA, R3                      ; 1416
                      52       12     A3   9E 0001A                  MOVAB    18(R3), R2                         ;
                      58              01   D0 0001E                  MOVL     #1, NOMSG                          ; 1428
                      62              04   8A 00021                  BICB2    #4, (R2)                           ; 1429
               10     AE       14     AE   9E 00024                  MOVAB    LIBRARYSTRING, LIBRARYNAME+4       ; 1430
                      10       18     AC   E8 00029                  BLBS     SUBPROMPTLEN, 1$                   ; 1432
       0A    0000'    CF              0B   E1 0002D                  BBC      #11, HELP_FLAGS, 1$                ; 1433
                               04     BC   B5 00033                  TSTW     @KEYDESCS                          ; 1434
                                      05   13 00036                  BEQL     1$                                 ;
                      62              02   8A 00038                  BICB2    #2, (R2)                           ; 1435
                                      03   11 0003B                  BRB      2$                                 ;
                      62              02   88 0003D 1$:              BISB2    #2, (R2)                           ; 1436
                               04     AC   DD 00040 2$:              PUSHL    KEYDESCS                           ; 1439
                                      53   DD 00043                  PUSHL    R3                                 ;
                               08     AC   DD 00045                  PUSHL    OUTPUTWIDTH                        ;
```

```
                                  57 DD 00048          PUSHL   R7                                       ; 1438
                     69           04 FB 0004A          CALLS   #4, CALL_LBRHELP                         ;
                     54           50 D0 0004D          MOVL    R0, STATUS                               ;
                     2C           54 E9 00050          BLBC    STATUS, 4$                               ;
      06             62           02 E1 00053          BBC     #2, (R2), 3$                             ; 1442
                     62           04 8A 00057          BICB2   #4, (R2)                                 ; 1444
                                00D7 31 0005A          BRW     16$                                      ; 1445
                     62           02 8A 0005D 3$:      BICB2   #2, (R2)                                 ; 1448
      1C      0000'  CF           02 E1 00060          BBC     #2, HELP_FLAGS+1, 5$                     ; 1450
                                  66 D5 00066          TSTL    (R6)                                     ; 1451
                                  18 19 00068          BLSS    5$                                       ;
              08     AE       10  BC D0 0006A          MOVL    @INDICES, CURRENT_LIBINDEX               ; 1453
                              04  AC DD 0006F          PUSHL   KEYDESCS                                 ; 1455
                                  53 DD 00072          PUSHL   R3                                       ;
                              08  AC DD 00074          PUSHL   OUTPUTWIDTH                              ;
                                  57 DD 00077          PUSHL   R7                                       ; 1454
                     69           04 FB 00079          CALLS   #4, CALL_LBRHELP                         ;
                     54           50 D0 0007C          MOVL    R0, STATUS                               ;
                     7E           54 E9 0007F 4$:      BLBC    STATUS, 11$                              ;
      05             62           02 E1 00082 5$:      BBC     #2, (R2), 6$                             ; 1459
                     55           02 CE 00086          MNEGL   #2, CURRENT_LIBNUMBER                    ; 1460
                                  03 11 00089          BRB     7$                                       ;
                     55           01 CE 0008B 6$:      MNEGL   #1, CURRENT_LIBNUMBER                    ; 1461
                     6E           01 CE 0008E 7$:      MNEGL   #1, USER_LIBNO                           ; 1463
      6F             62           02 E0 00091 8$:      BBS     #2, (R2), 12$                            ; 1464
                                  5E DD 00095          PUSHL   SP                                       ; 1465
                              08  AE 9F 00097          PUSHAB  DEFLIB_ACMODE                            ;
                              14  AE 9F 0009A          PUSHAB  LIBRARYNAME                              ;
              0000V  CF           03 FB 0009D          CALLS   #3, TRAN_NEXT_LIB                        ;
                                  50 D5 000A2          TSTL    R0                                       ;
                                  39 13 000A4          BEQL    10$                                      ;
                                  55 D6 000A6          INCL    CURRENT_LIBNUMBER                        ; 1467
                     66           55 D1 000A8          CMPL    CURRENT_LIBNUMBER, (R6)                  ; 1468
                                  E4 13 000AB          BEQL    8$                                       ;
                                  58 DD 000AD          PUSHL   NOMSG                                    ; 1469
                              10  AE 9F 000AF          PUSHAB  LIBRARYNAME                              ;
                              10  AE 9F 000B2          PUSHAB  CURRENT_LIBINDEX                         ;
              0000V  CF           03 FB 000B5          CALLS   #3, OPEN_LIBRARY                         ;
                     D4           50 E9 000BA          BLBC    R0, 8$                                   ;
                              04  AC DD 000BD          PUSHL   KEYDESCS                                 ; 1472
                                  53 DD 000C0          PUSHL   R3                                       ;
                              08  AC DD 000C2          PUSHL   OUTPUTWIDTH                              ;
                              14  AE 9F 000C5          PUSHAB  CURRENT_LIBINDEX                         ; 1471
                     69           04 FB 000C8          CALLS   #4, CALL_LBRHELP                         ;
                     54           50 D0 000CB          MOVL    R0, STATUS                               ;
      08             62           02 E0 000CE          BBS     #2, (R2), 9$                             ; 1473
                              08  AE 9F 000D2          PUSHAB  CURRENT_LIBINDEX                         ; 1474
              0000V  CF           01 FB 000D5          CALLS   #1, CLOSE_LIBRARY                        ;
                     B4           54 E8 000DA 9$:      BLBS    STATUS, 8$                               ; 1475
                                  21 11 000DD          BRB     11$                                      ;
      21             62           02 E0 000DF 10$:     BBS     #2, (R2), 12$                            ; 1479
              08     AE           67 D0 000E3          MOVL    (R7), CURRENT_LIBINDEX                   ; 1481
                     55           66 D0 000E7          MOVL    (R6), CURRENT_LIBNUMBER                  ; 1482
                     62           02 88 000EA          BISB2   #2, (R2)                                 ; 1483
                              04  AC DD 000ED          PUSHL   KEYDESCS                                 ; 1485
                                  53 DD 000F0          PUSHL   R3                                       ;
                              08  AC DD 000F2          PUSHL   OUTPUTWIDTH                              ;
```

```
                                   57  DD 000F5          PUSHL   R7                                  ; 1484
                       69          04  FB 000F7          CALLS   #4, CALL_LBRHELP                    ;
                       54          50  D0 000FA          MOVL    R0, STATUS                          ;
                       04          54  E8 000FD          BLBS    STATUS, 12$                         ;
                       50          54  D0 00100 11$:     MOVL    STATUS, R0                          ; 1486
                                       04 00103          RET                                         ;
                       66          55  D1 00104 12$:     CMPL    CURRENT_LIBNUMBER, (R6)             ; 1489
                                   2B  13 00107          BEQL    16$                                 ;
                                   66  D5 00109          TSTL    (R6)                                ; 1491
                                   07  19 0010B          BLSS    13$                                 ;
                                   57  DD 0010D          PUSHL   R7                                  ; 1492
                0000V  CF          01  FB 0010F          CALLS   #1, CLOSE_LIBRARY                   ;
                                   55  D5 00114 13$:     TSTL    CURRENT_LIBNUMBER                   ; 1493
                                   12  19 00116          BLSS    14$                                 ;
                            14     AC  DD 00118          PUSHL   LIBNAME                             ; 1496
                            10     AE  9F 0011B          PUSHAB  LIBRARYNAME                         ;
                0000V  CF          02  FB 0011E          CALLS   #2, SWITCH_LIBNAME                  ;
                0000'  CF          04  88 00123          BISB2   #4, PROMPT_FLAGS                    ; 1497
                                   03  11 00128          BRB     15$                                 ; 1493
                            14     BC  B4 0012A 14$:     CLRW    @LIBNAME                            ; 1500
                       67   08     AE  D0 0012D 15$:     MOVL    CURRENT_LIBINDEX, (R7)              ; 1501
                       66          55  D0 00131          MOVL    CURRENT_LIBNUMBER, (R6)             ; 1502
                       50          01  D0 00134 16$:     MOVL    #1, R0                              ; 1505
                                       04 00137          RET                                         ; 1506

; Routine Size:  312 bytes,    Routine Base:  $CODE$ + 052E
```

```
%SBTTL 'Routine change_lib';
ROUTINE change_lib (getcmddesc, charpos, indices, libname) =
BEGIN

!++
! FUNCTIONAL DESCRIPTION:
!
!       This routine changes the library context currently in effect to
!       the library specified by the command descriptor.  It also removes
!       the library specification from the command string.
!
! INPUTS:
!
!       getcmddesc =    address of descriptor containing the new library name
!
!       charpos =       pointer to "@" preceeding library name
!
!       indices =       address of data structure containing indices of
!                       libraries currently in use
!
!       libname =       address of descriptor for the user default library name
!
! OUTPUTS:
!
!       getcmddesc =    library specification is removed from string
!
!       indices =       if new library successfully found, updated to
!                       reflect library that was specified by the
!                       command string
!
!       libprompt =     if new library successfully found, updated to
!                       the file name of that library
!
! ROUTINE VALUE:
!
!       True, if new library found.
!       False, if library unchanged.
!--
MAP
    indices : REF BBLOCK,
    getcmddesc : REF BBLOCK;

BIND
    last_libindex = indices [hli$l_lastindex],          ! Index of last library examined
    last_libnumber = indices [hli$l_lastnumb];          ! No. of last library examined, relative to all defa

LOCAL
    current_libindex,                                   ! Index of library currently being searched
    current_libnumber,                                  ! Number of library currently being searched
    deflib_acmode : BYTE,                               ! Logical name table number
    librarystring : BBLOCK [nam$c_maxrss],              ! Default library name string
    libraryname : BBLOCK [dsc$c_s_bln],                 ! String descriptor for library name
    name_end,                                           ! End of a string
    name_len,                                           ! Length of a string
    nomsg,                                              ! Open_library message flag
    temp_end,                                           ! Location of first '/' in command string
    user_libno;                                         ! HLP$LIBRARY number
```

```
2
2 remove_terminator (.getcmddesc);                                            ! Remove command terminator
2
2 IF (.charpos + 1) EQL                                                       ! If no file name
3    (.getcmddesc [dsc$w_length] + .getcmddesc [dsc$a_pointer])
2    THEN RETURN false;                                                       ! Then return false
2
2 nomsg = true;                                                               ! Do not signal error if library can
2 libraryname [dsc$a_pointer] = librarystring;                                 ! Initialize local descriptor
2 make_upper_case (.getcmddesc, .getcmddesc [dsc$a_pointer]);                 ! Upper case command
2
2 name_len = .getcmddesc [dsc$w_length] +                                     ! Calculate length of @ to end of li
2            .getcmddesc [dsc$a_pointer] - .charpos;
2 IF (name_end = CH$FIND_CH (.name_len, .charpos, %C' ')) EQL 0                ! If no blank in command string
2    THEN name_end = .getcmddesc [dsc$w_length]                               ! Then end is end of command
2                 + .getcmddesc [dsc$a_pointer];
2 IF (temp_end = CH$FIND_CH (.name_len, .charpos, %C'/')) NEQ 0               ! If '/' position is before blank po
2    THEN IF .temp_end LSS .name_end
2            THEN name_end = .temp_end;                                       ! Then end  is position of '/'
2 name_len = .name_end - .charpos;                                            ! Calculate length of @ string
2
2 current_libnumber = -1;                                                     ! Init library number
2 user_libno = -1;                                                            ! Init default library searching
3 WHILE (tran_next_lib (libraryname, deflib_acmode, user_libno)
3         EQL true)
3 DO BEGIN
3    LOCAL
3         filename : BBLOCK [dsc$c_s_bln],
3         filebuffer : VECTOR [filename_length, BYTE];
3
3    filename = 0;
3    filename [dsc$a_pointer] = filebuffer;
3    switch_libname (libraryname, filename);
3
3    current_libnumber = .current_libnumber + 1;                              ! Incr libnumber
3
3    IF CH$EQL (.name_len - 1, .filename [dsc$a_pointer],                     ! Is command a substring of the file
3         .name_len - 1, .charpos + 1, 0)
3
4       THEN IF (open_library (current_libindex, libraryname, .nomsg))        ! If library successfully opened
4               THEN BEGIN
4                    IF .last_libnumber GEQ external_libnumber                ! If last library not main library
4                       THEN close_library (last_libindex);                   ! Then close it
4                    switch_libname (libraryname, .libname);                  ! Change library name
4                    last_libindex = .current_libindex;                       ! Set libindex
4                    last_libnumber = .current_libnumber;                     ! Set libnumber
4                    getcmddesc [dsc$w_length] =                              ! Remove @ string from command
4                         .getcmddesc [dsc$w_length] - .name_len;
4                    getcmddesc [dsc$a_pointer] =
4                         .getcmddesc [dsc$a_pointer] + .name_len;
4                    RETURN true;                                             ! Return success
3                    END;
3
2    END;                                                                     ! Library not found
2
2 !
2 ! If library file name was not found in logical name tables,
;   898         1621  2 ! then assume that the file name is actually a full file spec
;   899         1622  2 ! for a library that exists but is not a user-defined default
;   900         1623  2 ! library.  Try to open that library.
;   901         1624  2 !
;   902         1625  2
;   903         1626  2 libraryname [dsc$w_length] = .name_len - 1;                        ! Initialize library name
;   904         1627  2 libraryname [dsc$a_pointer] = .charpos + 1;
;   905         1628  2
;   906         1629  3 IF NOT (open_library (current_libindex, libraryname, .nomsg))      ! Try opening library
;   907         1630  2    THEN RETURN false;                                             ! If unsuccessful then give up and s
;   908         1631  2
;   909         1632  2 IF .last_libnumber GEQ external_libnumber                          ! If last library not main library
;   910         1633  2    THEN close_library (last_libindex);                            ! Then close it
;   911         1634  2 switch_libname (libraryname, .libname);                           ! Change libary name
;   912         1635  2 last_libindex = .current_libindex;                                ! Set libindex
;   913         1636  2 last_libnumber = -1;                                              ! Set libnumber
;   914         1637  2 getcmddesc [dsc$w_length] = .getcmddesc [dsc$w_length] - .name_len;   ! Remove a string from command
;   915         1638  2 getcmddesc [dsc$a_pointer] = .getcmddesc [dsc$a_pointer] + .name_len;
;   916         1639  2
;   917         1640  2 RETURN true;                                                      ! Return success
;   918         1641  1 END;                                                              ! Of change_lib



                                                 0FFC 00000 CHANGE_LIB:
                                                                 .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11       ; 1508
                           5E       FEBC    CE  9E 00002         MOVAB    -324(SP), SP                                ;
             5B      0C    AC               04  C1 00007         ADDL3    #4, INDICES, R11                            ; 1550
             5A      0C    AC               08  C1 0000C         ADDL3    #8, INDICES, R10                            ; 1551
                           55         04    AC  D0 00011         MOVL     GETCMDDESC, R5                              ; 1565
                                            55  DD 00015         PUSHL    R5                                          ;
                  0000V    CF               01  FB 00017         CALLS    #1, REMOVE_TERMINATOR                       ;
                           56         08    AC  D0 0001C         MOVL     CHARPOS, R6                                 ; 1567
                           51         01    A6  9E 00020         MOVAB    1(R6), R1                                   ;
                           50               65  3C 00024         MOVZWL   (R5), R0                                    ; 1568
                           50         04    A5  C0 00027         ADDL2    4(R5), R0                                   ;
                           50               51  D1 0002B         CMPL     R1, R0                                      ;
                                            03  12 0002E         BNEQ     1$                                          ;
                                          00F8  31 00030         BRW      11$                                         ;
                           59               01  D0 00033 1$:     MOVL     #1, NOMSG                                   ; 1571
                    40     AE         44    AE  9E 00036         MOVAB    LIBRARYSTRING, LIBRARYNAME+4                ; 1572
                                      04    A5  DD 0003B         PUSHL    4(R5)                                       ; 1573
                                            55  DD 0003E         PUSHL    R5                                          ;
                  0000V    CF               02  FB 00040         CALLS    #2, MAKE_UPPER_CASE                         ;
                           53               65  3C 00045         MOVZWL   (R5), R3                                    ; 1576
                           53         04    A5  C0 00048         ADDL2    4(R5), R3                                   ;
             54            53               56  C3 0004C         SUBL3    R6, R3, NAME_LEN                            ;
             66            54               20  3A 00050         LOCC     #32, NAME_LEN, (R6)                         ; 1577
                                            02  12 00054         BNEQ     2$                                          ;
                                            51  D4 00056         CLRL     R1                                          ;
                           52               51  D0 00058 2$:     MOVL     R1, NAME_END                                ;
                                            03  12 0005B         BNEQ     3$                                          ;
                           52               53  D0 0005D         MOVL     R3, NAME_END                                ; 1579
             66            54               2F  3A 00060 3$:     LOCC     #47, NAME_LEN, (R6)                         ; 1580
                                            02  12 00064         BNEQ     4$                                          ;
                                          51  D4 00066        CLRL     R1                                           ;
                                          51  D5 00068 4$:    TSTL     TEMP_END                                     ;
                                          08  13 0006A        BEQL     5$                                           ;
                        52                51  D1 0006C        CMPL     TEMP_END, NAME_END                           ; 1581
                                          03  18 0006F        BGEQ     5$                                           ;
                        52                51  D0 00071        MOVL     TEMP_END, NAME_END                           ; 1582
            54          52                56  C3 00074 5$:    SUBL3    R6, NAME_END, NAME_LEN                       ; 1583
                        58                01  CE 00078        MNEGL    #1, CURRENT_LIBNUMBER                        ; 1585
                        6E                01  CE 0007B        MNEGL    #1, USER_LIBNO                               ; 1586
                        57         FF     A4  9E 0007E        MOVAB    -1(R4), R7                                   ; 1600
                                          5E  DD 00082 6$:    PUSHL    SP                                           ; 1587
                                   08     AE  9F 00084        PUSHAB   DEFLIB_ACMODE                                ;
                                   44     AE  9F 00087        PUSHAB   LIBRARYNAME                                  ;
                  0000V   CF              03  FB 0008A        CALLS    #3, TRAN_NEXT_LIB                            ;
                          01              50  D1 0008F        CMPL     R0, #1                                       ; 1588
                                          51  12 00092        BNEQ     8$                                           ;
                                   34     AE  D4 00094        CLRL     FILENAME                                     ; 1594
                     38   AE       0C     AE  9E 00097        MOVAB    FILEBUFFER, FILENAME+4                       ; 1595
                                   34     AE  9F 0009C        PUSHAB   FILENAME                                     ; 1596
                                   40     AE  9F 0009F        PUSHAB   LIBRARYNAME                                  ;
                  C000V   CF              02  FB 000A2        CALLS    #2, SWITCH_LIBNAME                           ;
                                          58  D6 000A7        INCL     CURRENT_LIBNUMBER                            ; 1598
       01   A6       38   BE              57  29 000A9        CMPC3    R7, @FILENAME+4, 1(R6)                       ; 1600
                                          D1  12 000AF        BNEQ     6$                                           ;
                                          59  DD 000B1        PUSHL    NOMSG                                        ; 1603
                                   40     AE  9F 000B3        PUSHAB   LIBRARYNAME                                  ;
                                   10     AE  9F 000B6        PUSHAB   CURRENT_LIBINDEX                             ;
                  0000V   CF              03  FB 000B9        CALLS    #3, OPEN_LIBRARY                             ;
                          C1              50  E9 000BE        BLBC     R0, 6$                                       ;
               FFFFFFFF   8F              6A  D1 000C1        CMPL     (R10), #-1                                   ; 1605
                                          07  19 000C8        BLSS     7$                                           ;
                                          5B  DD 000CA        PUSHL    R11                                          ; 1606
                  0000V   CF              01  FB 000CC        CALLS    #1, CLOSE_LIBRARY                            ;
                                   10     AC  DD 000D1 7$:    PUSHL    LIBNAME                                      ; 1607
                                   40     AE  9F 000D4        PUSHAB   LIBRARYNAME                                  ;
                  0000V   CF              02  FB 000D7        CALLS    #2, SWITCH_LIBNAME                           ;
                          6B       08     AE  D0 000DC        MOVL     CURRENT_LIBINDEX, (R11)                      ; 1608
                          6A              58  D0 000E0        MOVL     CURRENT_LIBNUMBER, (R10)                     ; 1609
                                          3B  11 000E3        BRB      10$                                          ; 1611
                     3C   AE              57  B0 000E5 8$:    MOVW     R7, LIBRARYNAME                              ; 1626
                     40   AE       01     A6  9E 000E9        MOVAB    1(R6), LIBRARYNAME+4                         ; 1627
                                          59  DD 000EE        PUSHL    NOMSG                                        ; 1629
                                   40     AE  9F 000F0        PUSHAB   LIBRARYNAME                                  ;
                                   10     AE  9F 000F3        PUSHAB   CURRENT_LIBINDEX                             ;
                  0000V   CF              03  FB 000F6        CALLS    #3, OPEN_LIBRARY                             ;
                          2D              50  E9 000FB        BLBC     R0, 11$                                      ;
               FFFFFFFF   8F              6A  D1 000FE        CMPL     (R10), #-1                                   ; 1632
                                          07  19 00105        BLSS     9$                                           ;
                                          5B  DD 00107        PUSHL    R11                                          ; 1633
                  0000V   CF              01  FB 00109        CALLS    #1, CLOSE_LIBRARY                            ;
                                   10     AC  DD 0010E 9$:    PUSHL    LIBNAME                                      ; 1634
                                   40     AE  9F 00111        PUSHAB   LIBRARYNAME                                  ;
                  0000V   CF              02  FB 00114        CALLS    #2, SWITCH_LIBNAME                           ;
                          6B       08     AE  D0 00119        MOVL     CURRENT_LIBINDEX, (R11)                      ; 1635
                          6A              01  CE 0011D        MNEGL    #1, (R10)                                    ; 1636
                          65              54  A2 00120 10$:   SUBW2    NAME_LEN, (R5)                               ; 1637
                     04   A5              54  C0 00123        ADDL2    NAME_LEN, 4(R5)                              ; 1638
```

```
                          50             01  D0 00127          MOVL    #1, R0                                          ; 1640
                                             04 0012A          RET                                                     ;
                                         50  D4 0012B 11$:     CLRL    R0                                              ; 1641
                                             04 0012D          RET                                                     ;

; Routine Size:  302 bytes,    Routine Base:  $CODE$ + 0666
```

```
  1 %SBTTL 'Routine switch_libname';
  1 ROUTINE switch_libname (newname, oldname) =
  2 BEGIN
  2
  2 !++
  2 ! FUNCTIONAL DESCRIPTION:
  2 !
  2 !       This routine inserts a new libname into the old descriptor.
  2 !
  2 ! INPUTS:
  2 !
  2 !       newname =       address of a descriptor for the new library file name
  2 !
  2 !       oldname =       address of a descriptor for the old library file name
  2 !
  2 ! OUTPUTS:
  2 !
  2 !       oldname =       updated to specify the name of the new library
  2 !
  2 ! ROUTINE VALUE:
  2 !
  2 !       Always true.
  2 !
  2 !--
  2 MAP
  2     newname : REF BBLOCK,
  2     oldname : REF BBLOCK;
  2
  2 LOCAL
  2     rsabuf: BBLOCK [nam$c_maxrss],      ! buffer for resultant string
  2     esabuf: BBLOCK [nam$c_maxrss],      ! buffer for expanded string
  2     libfab: BBLOCK [fab$c_bln],         ! temporary FAB
  2     libnam: BBLOCK [nam$c_bln];         ! temporary NAM block
  2
P 2 $NAM_INIT( NAM=libnam,
P 2             RSA=rsabuf,
P 2             RSS=nam$c_maxrss,
P 2             ESA=esabuf,
  2             ESS=nam$c_maxrss);
  2
P 2 $FAB_INIT( FAB=libfab,
P 2             FOP=NAM,
P 2             FNA=.newname[dsc$a_pointer],
P 2             FNS=.newname[dsc$w_length],
  2             NAM=libnam);
  2
  2 $PARSE(FAB=libfab);
  2
  2 oldname[dsc$w_length] = .libnam[nam$b_name];
  2 CH$MOVE(.libnam[nam$b_name], .libnam[nam$l_name], .oldname[dsc$a_pointer]);
  2
  2 RETURN true;
  1 END;                                                  ! Of switch_libname


                                                 .EXTRN  SYS$PARSE
```

```
                                                       003C 00000 SWITCH_LIBNAME:
                                                                        .WORD   Save R2,R3,R4,R5                            ; 1643
                                     5E      FD50   CE   9E 00002       MOVAB   -688(SP), SP
   0060   8F          00             6E             00   2C 00007       MOVC5   #0, (SP), #0, #96, $RMS_PTR                 ; 1680
                                                    6E      0000E
                                     6E      6002   8F   B0 0000F       MOVW    #24578, $RMS_PTR
                                02   AE             01   8E 00014       MNEGB   #1, $RMS_PTR+2
                                04   AE      FF00   CD   9E 00018       MOVAB   RSABUF, $RMS_PTR+4
                                0A   AE             01   8E 0001E       MNEGB   #1, $RMS_PTR+10
                                0C   AE      00B0   CE   9E 00022       MOVAB   ESABUF, $RMS_PTR+12
   0050   8F          00             6E             00   2C 00028       MOVC5   #0, (SP), #0, #80, $RMS_PTR                 ; 1686
                                             60     AE      0002F
                                60   AE      5003   8F   B0 00031       MOVW    #20483, $RMS_PTR
                                64   AE  01000000   8F   D0 00037       MOVL    #16777216, $RMS_PTR+4
                                76   AE             02   90 0003F       MOVB    #2, $RMS_PTR+22
                                7F   AE             02   90 00043       MOVB    #2, $RMS_PTR+31
                              0088   CE             6E   9E 00047       MOVAB   LIBNAM, $RMS_PTR+40
                                     50        04   AC   D0 0004C       MOVL    NEWNAME, R0
                              008C   CE        04   A0   D0 00050       MOVL    4(R0), $RMS_PTR+44
                              0094   CE             60   90 00056       MOVB    (R0), $RMS_PTR+52
                                               60   AE   9F 0005B       PUSHAB  LIBFAB                                      ; 1688
                         00000000G   00             01   FB 0005E       CALLS   #1, SYS$PARSE
                                     50        08   AC   D0 00065       MOVL    OLDNAME, R0                                 ; 1690
                                     60        3B   AE   9B 00069       MOVZBW  LIBNAM+59, (R0)
                                     51        3B   AE   9A 0006D       MOVZBL  LIBNAM+59, R1                               ; 1691
              04      B0        4C   BE             51   28 00071       MOVC3   R1, @LIBNAM+76, @4(R0)
                                     50             01   D0 00077       MOVL    #1, R0                                      ; 1693
                                                         04 0007A       RET                                                 ; 1694

; Routine Size:  123 bytes,    Routine Base:  $CODE$ + 0794
```

```
1 %SBTTL 'Routine tran_next_lib';
1 ROUTINE tran_next_lib (libname, deflibacmode, userlibno) =
2 BEGIN
2
2 !++
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       This routine returns the value true if a default
2 !       library if found and false if not.  If a library is found,
2 !       the library file name is returned in the descriptor libname.
2 !
2 ! INPUTS:
2 !
2 !       libname =       address of a descriptor to return the library name in
2 !
2 !       deflibacmode =  address of a longword containing the access mode of the
2 !                       last logical name table searched
2 !
2 !       userlibno =     address of a longword containg the number of the last
2 !                       library found in the last logical name table.  If the
2 !                       number is -1, then start searching the logical name
2 !                       tables all over again.
2 !
2 ! OUTPUTS:
2 !
2 !       libname, deflibacmode, userlibno : as described above
2 !
2 ! ROUTINE VALUE:
2 !
2 !       True, if user default library is found.
2 !       False, if no more user default libraries left, i.e., at end of tables.
2 !
2 !--
2
2 MAP
2     help_flags : BITVECTOR,
2     libname : REF BBLOCK;
2
2 BIND
2     maxliblen = %CHARCOUNT('HLP$LIBRARY_999');          ! Max lib logical name length
2
2 OWN
2     hlplibnam : COUNTEDSTRING ('HLP$LIBRARY'),          ! Initial library logical name
2     libnamefao : COUNTEDSTRING ('HLP$LIBRARY_!UW');     ! FAO descriptor for general library logical name
2
2 LOCAL
2     deflibdsbmsk : VECTOR [4,BYTE],                     ! Logical name table access modes
2     deflibnam : VECTOR [maxliblen,BYTE],                ! Library logical name buffer
2     deflibdesc : BBLOCK [dsc$c_s_bln],                  ! Library logical name descriptor
2     status;                                             ! Status of logical name translation
2
2 IF ..userlibno LSS 0                                                ! If starting search at beginning
3    THEN BEGIN                                                       ! Then initialize access mode and li
3         .deflibacmode = 0;
3         .userlibno = 0;
2         END;
2
```

```
  2 deflibdesc [dsc$w_length] = maxliblen;                                   ! Initialize logical name descriptor
  2 deflibdesc [dsc$a_pointer] = deflibnam;
  2
  2 IF ..userlibno EQL 0                                                     ! If first logical name in table
  3    THEN BEGIN                                                            ! Then special case, do not use FAO
  3         deflibdesc [dsc$w_length] = .hlplibnam [0];
  3         CH$MOVE (.deflibdesc [dsc$w_length], hlplibnam + 1, deflibnam);
  3         END
  3
  3    ELSE BEGIN                                                            ! Else use FAO to combine logical na
  3         deflibdesc [dsc$w_length] = .libnamefao [0];
  3         CH$MOVE (.deflibdesc [dsc$w_length], libnamefao + 1, deflibnam);
  4         IF NOT $FAO (deflibdesc, deflibdesc, deflibdesc, ..userlibno)
  3            THEN RETURN false;
  2         END;
  2
  2 IF .help_flags [..deflibacmode + 1] NEQ 0                                ! If searching of current logical na
  3    THEN BEGIN                                                            ! Then translate logical name for th
  3         deflibdsbmsk = %X'00060503';
  3         libname [dsc$w_length] = nam$c_maxrss;
P 3         status = $TRNLOG (LOGNAM = deflibdesc,
P 3                           DSBMSK = .deflibdsbmsk [..deflibacmode],
P 3                           RSLLEN = .libname,
  3                           RSLBUF = .libname);
  4         IF .status AND (.status NEQ SS$_NOTRAN)                          ! If logical name is successfully tr
  4            THEN BEGIN
  4                 .userlibno = ..userlibno + 1;                            ! Then increment lib no. for next se
  4                 RETURN true;                                             ! And return true
  3                 END;
  2         END;
  2
  2 !
  2 ! If current logical name table not enabled or logical name unsuccessfully
  2 ! translated, then recursively call this routine to search next table.
  2 !
  2
  2 IF (.deflibacmode = ..deflibacmode + 1) GTR 2                            ! Increment access mode
  2    THEN RETURN false;                                                    ! If out of tables, then return fals
  2 .userlibno = 0;                                                          ! Reset lib no. to start of table
  2 RETURN tran_next_lib (.libname, .deflibacmode, .userlibno);              ! Make recursive call
  2
  1 END;                                                                     ! Of tran_next_lib


                                                            .PSECT  $OWN$,NOEXE,2

                                               0B  00038 HLPLIBNAM:
                                                            .BYTE   11
                 59 52 41 52 42 49 4C 24 50 4C 48  00039    .ASCII  \HLP$LIBRARY\                      ;
                                               0F  00044 LIBNAMEFAO:
                                                            .BYTE   15                                 ;
     57 55 21 5F 59 52 41 52 42 49 4C 24 50 4C 48  00045    .ASCII  \HLP$LIBRARY_!UW\                  ;

                                                      MAXLIBLEN=      15
                                                            .EXTRN  SYS$FAO, SYS$TRNLOG
```

```
                                                     E 15
LBR_OUTPUTHELP  Prompting and library searching help function   16-Sep-1984 02:04:00   VAX-11 Bliss-32 V4.0-742          Page  34
V04-000         Routine tran_next_lib                           14-Sep-1984 12:37:45   [LBR.SRC]OUTPUTHLP.B32;1                (8)

                                                                .PSECT  $CODE$,NOWRT,2

                                            00FC 00000 TRAN_NEXT_LIB:
                                                                .WORD   Save R2,R3,R4,R5,R6,R7                    ; 1696
                            57      0000'    CF 9E 00002        MOVAB   HLPLIBNAM, R7                             ;
                            5E               1C C2 00007        SUBL2   #28, SP                                   ;
                            56         0C    AC D0 0000A        MOVL    USERLIBNO, R6                             ; 1746
                                             66 D5 0000E        TSTL    (R6)                                      ;
                                             05 18 00010        BGEQ    1$                                        ;
                                       08    BC D4 00012        CLRL    @DEFLIBACMODE                             ; 1748
                                             66 D4 00015        CLRL    (R6)                                      ; 1749
                       04   AE               0F B0 00017 1$:    MOVW    #15, DEFLIBDESC                           ; 1752
                       08   AE         0C    AE 9E 0001B        MOVAB   DEFLIBNAM, DEFLIBDESC+4                   ; 1753
                                             66 D5 00020        TSTL    (R6)                                      ; 1755
                                             0D 12 00022        BNEQ    2$                                        ;
                       04   AE               67 9B 00024        MOVZBW  HLPLIBNAM, DEFLIBDESC                     ; 1757
      0C    AE         01   A7         04    AE 28 00028        MOVC3   DEFLIBDESC, HLPLIBNAM+1, DEFLIBNAM        ; 1758
                                             21 11 0002F        BRB     3$                                        ; 1755
                       04   AE         0C    A7 9B 00031 2$:    MOVZBW  LIBNAMEFAO, DEFLIBDESC                    ; 1762
      0C    AE         0D   A7         04    AE 28 00036        MOVC3   DEFLIBDESC, LIBNAMEFAO+1, DEFLIBNAM       ; 1763
                                             66 DD 0003D        PUSHL   (R6)                                      ; 1764
                                       08    AE 9F 0003F        PUSHAB  DEFLIBDESC                                ;
                                       0C    AE 9F 00042        PUSHAB  DEFLIBDESC                                ;
                                       10    AE 9F 00045        PUSHAB  DEFLIBDESC                                ;
                00000000G   00               04 FB 00048        CALLS   #4, SYS$FAO                               ;
                            60               50 E9 0004F        BLBC    R0, 5$                                    ;
                            50         08    BC D0 00052 3$:    MOVL    @DEFLIBACMODE, R0                         ; 1768
                            52         01    A0 9E 00056        MOVAB   1(R0), R2                                 ;
51    F8    A7              01               52 EF 0005A        EXTZV   R2, #1, HELP_FLAGS, R1                    ;
                                             34 13 00060        BEQL    4$                                        ;
                            6E   00060503    8F D0 00062        MOVL    #394499, DEFLIBDSBMSK                     ; 1770
                       04   BC         FF    8F 9B 00069        MOVZBW  #255, @LIBNAME                            ; 1771
                            7E             6E40 9A 0006E        MOVZBL  DEFLIBDSBMSK[R0], -(SP)                   ; 1775
                                             7E 7C 00072        CLRQ    -(SP)                                     ;
                                       04    AC DD 00074        PUSHL   LIBNAME                                   ;
                                       04    AC DD 00077        PUSHL   LIBNAME                                   ;
                                       18    AE 9F 0007A        PUSHAB  DEFLIBDESC                                ;
                00000000G   00               06 FB 0007D        CALLS   #6, SYS$TRNLOG                            ;
                            0F               50 E9 00084        BLBC    STATUS, 4$                                ; 1776
                00000629    8F               50 D1 00087        CMPL    STATUS, #1577                             ;
                                             06 13 0008E        BEQL    4$                                        ;
                                             66 D6 00090        INCL    (R6)                                      ; 1778
                            50               01 D0 00092        MOVL    #1, R0                                    ; 1779
                                                04 00095        RET                                               ;
            50         08   BC               01 C1 00096 4$:    ADDL3   #1, @DEFLIBACMODE, R0                     ; 1788
                       08   BC               50 D0 0009B        MOVL    R0, @DEFLIBACMODE                         ;
                            02               50 D1 0009F        CMPL    R0, #2                                    ;
                                             0E 14 000A2        BGTR    5$                                        ;
                                             66 D4 000A4        CLRL    (R6)                                      ; 1790
                                             56 DD 000A6        PUSHL   R6                                        ; 1791
                            7E         04    AC 7D 000A8        MOVQ    LIBNAME, -(SP)                            ;
                     FF4F   CF               03 FB 000AC        CALLS   #3, TRAN_NEXT_LIB                         ;
                                                04 000B1        RET                                               ;
                                             50 D4 000B2 5$:    CLRL    R0                                        ; 1793
                                                04 000B4        RET                                               ;

; Routine Size:  181 bytes,    Routine Base:  $CODE$ + 080F
```

```
  1 %SBTTL 'Routine open_library';
  1 ROUTINE open_library (libindex, libname, nomsgflag) =
  2 BEGIN
  2
  2 !++
  2 ! FUNCTIONAL DESCRIPTION:
  2 !
  2 !       Open the library with the specified name and return its index.
  2 !
  2 ! INPUTS:
  2 !
  2 !       libindex =      address of longword to contain index of opened library
  2 !
  2 !       libname =       address of descriptor for library name
  2 !
  2 !       nomsgflag =     flag that is true if open errors should not be signalled
  2 !
  2 ! OUTPUTS:
  2 !
  2 !       libindex : as described above
  2 !
  2 ! ROUTINE VALUE:
  2 !
  2 !       True, if library successfully opened.
  2 !       False, if unsuccessful.
  2 !
  2 !--
  2
  2 MAP
  2     libname : REF BBLOCK;
  2
  2 EXTERNAL
  2     lbr$gl_rmsstv : ADDRESSING_MODE (GENERAL);          ! RMS STV from librarian
  2
  2 LOCAL
  2     filnamdesc : BBLOCK [dsc$c_s_bln],                  ! Library name descriptor
  2     help_defname : BBLOCK [dsc$c_s_bln],                ! Default filename descriptor
  2     helpfilename : BBLOCK [nam$c_maxrss],               ! Resultant file name
  2     help_func,                                          ! Library access type
  2     help_type,                                          ! Type of library
  2     namblk : BBLOCK [nam$c_bln],                        ! Library name block
  2     status;
  2
P 2 $NAM_INIT (NAM = namblk,                                ! Initialize the NAM block
P 2         RSS = nam$c_maxrss,
P 2         RSA = helpfilename,
P 2         ESS = nam$c_maxrss,
  2         ESA = helpfilename);
  2
  2 help_func = lbr$c_read;                                 ! Will be reading the library
  2 help_type = lbr$c_typ_hlp;                              ! Library is of help type
  2
  2 !
  2 ! Call librarian to initialize control.  Stop if error.
  2 !
  2
  3 IF NOT (status = lbr$ini_control (.libindex, help_func, help_type, namblk))
```

```
; 1131          1851  2 THEN
; 1132          1852  3     BEGIN
; 1133          1853  3     SIGNAL (.status);
; 1134          1854  3     RETURN .status OR sts$m_inhib_msg;
; 1135          1855  2     END;
; 1136          1856  2
; 1137          1857  2 !
; 1138          1858  2 ! Call librarian to open library.  Stop if error should be flagged.
; 1139          1859  2 !
; 1140          1860  2
; 1141          1861  2 help_defname [dsc$w_length] = .syshelp [0];          ! Initialize default library directory and type
; 1142          1862  2 help_defname [dsc$a_pointer] = syshelp [1];
; 1143          1863  2
; 1144          1864  3 IF NOT (status = lbr$open (.libindex, .libname, 0, help_defname))
; 1145          1865  2    AND NOT .nomsgflag
; 1146          1866  3    THEN BEGIN
; 1147          1867  3         IF (filnamdesc [dsc$w_length] = .namblk [nam$b_esl]) NEQ 0
; 1148          1868  3            THEN filnamdesc [dsc$a_pointer] = .namblk [nam$l_esa]
; 1149          1869  4            ELSE BEGIN
; 1150          1870  4                 filnamdesc [dsc$w_length] = .libname [dsc$w_length];
; 1151          1871  4                 filnamdesc [dsc$a_pointer] = .libname [dsc$a_pointer];
; 1152          1872  3                 END;
; 1153          1873  3         SIGNAL (shr$_openin OR hlp$c_facility OR sts$k_error,
; 1154          1874  3                     1, filnamdesc, .status, .lbr$gl_rmsstv);
; 1155          1875  3         status = shr$_openin OR hlp$c_facility OR sts$k_error OR sts$m_inhib_msg;
; 1156          1876  2         END;
; 1157          1877  2
; 1158          1878  2 RETURN .status;
; 1159          1879  2
; 1160          1880  1 END;                                                     !Of open_library


                                                            .EXTRN  LBR$GL_RMSSTV

                                              007C 00000 OPEN_LIBRARY:
                                                            .WORD   Save R2,R3,R4,R5,R6                         ; 1795
                            56 00000000G  00  9E 00002      MOVAB   LIB$SIGNAL, R6                              ;
                            5E      FE88  CE  9E 00009      MOVAB   -376(SP), SP                                ;
   0060   8F          00    6E            00  2C 0000E      MOVC5   #0, (SP), #0, #96, $RMS_PTR                 ; 1841
                                      08  AE     00015                                                          ;
                      08    AE      6002  8F  B0 00017      MOVW    #24578, $RMS_PTR                            ;
                      0A    AE            01  8E 0001D      MNEGB   #1, $RMS_PTR+2                              ;
                      0C    AE        68  AE  9E 00021      MOVAB   HELPFILENAME, $RMS_PTR+4                    ;
                      12    AE            01  8E 00026      MNEGB   #1, $RMS_PTR+10                             ;
                      14    AE        68  AE  9E 0002A      MOVAB   HELPFILENAME, $RMS_PTR+12                   ;
                      04    AE            01  D0 0002F      MOVL    #1, HELP_FUNC                               ; 1843
                            6E            03  D0 00033      MOVL    #3, HELP_TYPE                               ; 1844
                                      08  AE  9F 00036      PUSHAB  NAMBLK                                      ; 1850
                                      04  AE  9F 00039      PUSHAB  HELP_TYPE                                   ;
                                      0C  AE  9F 0003C      PUSHAB  HELP_FUNC                                   ;
                                      04  AC  DD 0003F      PUSHL   LIBINDEX                                    ;
                 00000000G  00            04  FB 00042      CALLS   #4, LBR$INI_CONTROL                         ;
                            53            50  D0 00049      MOVL    R0, STATUS                                  ;
                            0E            53  E8 0004C      BLBS    STATUS, 1$                                  ;
                                          53  DD 0004F      PUSHL   STATUS                                      ; 1853
                            66            01  FB 00051      CALLS   #1, LIB$SIGNAL                              ;
```

```
                  50              53 10000000   8F   C9 00054          BISL3   #268435456, STATUS, R0              ; 1854
                                                     04 0005C          RET                                         ;
                          F0      AD     0000'  CF   9B 0005D 1$:      MOVZBW  SYSHELP, HELP_DEFNAME               ; 1861
                          F4      AD     0000'  CF   9E 00063          MOVAB   SYSHELP+1, HELP_DEFNAME+4           ; 1862
                                           F0   AD   9F 00069          PUSHAB  HELP_DEFNAME                        ; 1864
                                                7E   D4 0006C          CLRL    -(SP)                               ;
                                  52       08   AC   D0 0006E          MOVL    LIBNAME, R2                         ;
                                                52   DD 00072          PUSHL   R2                                  ;
                                           04   AC   DD 00074          PUSHL   LIBINDEX                            ;
                     00000000G    00            04   FB 00077          CALLS   #4, LBR$OPEN                        ;
                                  53            50   D0 0007E          MOVL    R0, STATUS                          ;
                                  38            53   E8 00081          BLBS    STATUS, 4$                          ;
                                  34       0C   AC   E8 00084          BLBS    NOMSGFLAG, 4$                       ; 1865
                          F8      AD       13   AE   9B 00088          MOVZBW  NAMBLK+11, FILNAMDESC               ; 1867
                                                07   13 0008D          BEQL    2$                                  ;
                          FC      AD       14   AE   D0 0008F          MOVL    NAMBLK+12, FILNAMDESC+4             ; 1868
                                                09   11 00094          BRB     3$                                  ;
                          F8      AD            62   B0 00096 2$:      MOVW    (R2), FILNAMDESC                    ; 1870
                          FC      AD       04   A2   D0 0009A          MOVL    4(R2), FILNAMDESC+4                 ; 1871
                                     00000000G  00   DD 0009F 3$:      PUSHL   LBR$GL_RMSSTV                       ; 1874
                                                53   DD 000A5          PUSHL   STATUS                              ;
                                           F8   AD   9F 000A7          PUSHAB  FILNAMDESC                          ; 1873
                                                01   DD 000AA          PUSHL   #1                                  ;
                                     0076109A   8F   DD 000AC          PUSHL   #7737498                            ;
                                  66            05   FB 000B2          CALLS   #5, LIB$SIGNAL                      ;
                                  53 1076109A   8F   D0 000B5          MOVL    #276172954, STATUS                  ; 1875
                                  50            53   D0 000BC 4$:      MOVL    STATUS, R0                          ; 1878
                                                04      000BF          RET                                         ; 1880

; Routine Size:  192 bytes,    Routine Base:  $CODE$ + 08C4
; 1162          1881  1 %SBTTL 'Routine close_library';
; 1163          1882  1 ROUTINE close_library (libindex) =
; 1164          1883  2 BEGIN
; 1165          1884  2
; 1166          1885  2 !++
; 1167          1886  2 ! FUNCTIONAL DESCRIPTION:
; 1168          1887  2 !
; 1169          1888  2 !       Close the open help library.
; 1170          1889  2 !
; 1171          1890  2 ! INPUTS:
; 1172          1891  2 !
; 1173          1892  2 !       libindex =      address of longword containing index of library to close.
; 1174          1893  2 !
; 1175          1894  2 ! OUTPUTS:
; 1176          1895  2 !
; 1177          1896  2 !       None.
; 1178          1897  2 !
; 1179          1898  2 ! ROUTINE VALUE:
; 1180          1899  2 !
; 1181          1900  2 !       Always true.
; 1182          1901  2 !
; 1183          1902  2 !--
; 1184          1903  2
; 1185          1904  2 LOCAL
; 1186          1905  2     status;
; 1187          1906  2
; 1188          1907  3 IF NOT (status = lbr$close (.libindex))
; 1189          1908  2    THEN SIGNAL (.status);
; 1190          1909  2
; 1191          1910  2 RETURN true
; 1192          1911  2
; 1193          1912  1 END;                                                    !Of close_library



                                            0000 00000 CLOSE_LIBRARY:
                                                                  .WORD    Save nothing                                   ; 1882
                                    04    AC   DD 00002           PUSHL    LIBINDEX                                       ; 1907
                     00000000G  00          01   FB 00005           CALLS    #1, LBR$CLOSE                                  ;
                                09          50   E8 0000C           BLBS     STATUS, 1$                                     ;
                                            50   DD 0000F           PUSHL    STATUS                                         ; 1908
                     00000000G  00          01   FB 00011           CALLS    #1, LIB$SIGNAL                                 ;
                                50          01   D0 00018 1$:       MOVL     #1, R0                                         ; 1910
                                                 04 0001B           RET                                                     ; 1912

; Routine Size:  28 bytes,    Routine Base:  $CODE$ + 0984
```

```
1 %SBTTL 'Routine setup_keys';
1 ROUTINE setup_keys (getcmddesc, keydescs, truekeys) =
2 BEGIN
2
2 !++
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !      This routine divides the descriptor for the line of keys into
2 !      individual descriptors for the individual keys.
2 !
2 ! INPUTS:
2 !
2 !      getcmddesc =    address of descriptor for the set of keys currently being
2 !                      processed
2 !
2 !      keydescs =      address of vector of key descriptors to be returned
2 !
2 !      truekeys =      address of longword to contain the number of keys found
2 !
2 ! OUTPUTS:
2 !
2 !      keydescs, turekeys : as described above
2 !
2 ! ROUTINE VALUE:
2 !
2 !--
2
2 MAP
2     truekeys : REF VECTOR [,BYTE],
2     getcmddesc : REF BBLOCK,
2     keydescs : REF VECTOR [,BYTE];
2
2 LOCAL
2     paren,                  ! Pointer to a left parenthesis
2     parm_begin,             ! Pointer to beginning of current key
2     parm_end,               ! Pointer to end of current key
2     next_qual,              ! Pointer to next qualifier
2     line_end;               ! Pointer to end of keys
2
2 paren = %ASCII '(';                                                     ! Init paren ptr
2 CH$FILL (0, dsc$c_s_bln * hlp$c_maxkeys, .keydescs);                    ! Clear key descriptors
2
2 !
2 ! Convert command line to upper case.  Then find the start of the help
2 ! keys, and fill in descriptors for them.
2 !
2
2 make_upper_case (.getcmddesc, .getcmddesc [dsc$a_pointer]);             ! Convert command line to upper case
2 parm_end = .getcmddesc [dsc$a_pointer];                                 ! Initialize pointers
2 line_end = .getcmddesc [dsc$a_pointer] + .getcmddesc [dsc$w_length];
2 next_qual = CH$FIND_CH (.getcmddesc [dsc$w_length],
2                         .getcmddesc [dsc$a_pointer], %ASCII '/');
2
2 INCRU level FROM 0 TO hlp$c_maxkeys - 1                                 ! Loop to search for keys
3 DO BEGIN
3
4    DO BEGIN                                                             ! Find first key that doesn't start
```

```
; 1252        1970  5      IF (parm_begin = CH$FIND_NOT_CH (.line_end - .parm_end,               ! Find start of key
; 1253        1971  4                               .parm_end, %ASCII ' ')) EQL 0
; 1254        1972  4         THEN parm_begin = .line_end;                                        ! Set right if none found
; 1255        1973  4
; 1256        1974  4      IF .next_qual NEQ 0                                                    ! If there is a qualifier
; 1257        1975  4         AND .parm_begin GEQU .next_qual                                     ! and qualifier is closer
; 1258        1976  5         THEN BEGIN
; 1259        1977  5              parm_begin = .next_qual;                                       ! then pick up the qualifier
; 1260        1978  5              next_qual = CH$FIND_CH (.line_end - .parm_begin - 1,           ! and find the next one
; 1261        1979  5                           .parm_begin + 1, %ASCII '/');
; 1262        1980  4              END;
; 1263        1981  4
; 1264        1982  5      IF (parm_end = CH$FIND_CH (.line_end - .parm_begin,                    ! Find end of key
; 1265        1983  4                               .parm_begin, %ASCII ' ')) EQL 0
; 1266        1984  4         THEN parm_end = .line_end;
; 1267        1985  4
; 1268        1986  4      IF .next_qual NEQ 0                                                    ! If a qualifier on line
; 1269        1987  4         AND .parm_end GTRU .next_qual                                       ! and qualifier is closer
; 1270        1988  4         THEN parm_end = .next_qual;                                         ! then it marks end of current param
; 1271        1989  4
; 1272        1990  4      END                                                                    ! Of until loop
; 1273        1991  4
; 1274        1992  4      UNTIL (.parm_end - .parm_begin EQL 0                                   ! No more keys
; 1275        1993  3            OR CH$NEQ (1, .parm_begin, 1, paren));                           ! or, key that starts with non-(
; 1276        1994  3
; 1277        1995  3 !
; 1278        1996  3 ! Fill in string descriptor
; 1279        1997  3 !
; 1280        1998  3
; 1281        1999  4       BEGIN
; 1282        2000  4
; 1283        2001  4           BIND
; 1284        2002  4               curkeydesc = keydescs [dsc$c_s_bln * .level] : BBLOCK;      ! Descriptro for current key
; 1285        2003  4
; 1286        2004  4           IF (curkeydesc [dsc$w_length] = .parm_end - .parm_begin) EQL 0  ! If key length is zero
; 1287        2005  5              THEN BEGIN
; 1288        2006  5                   IF .level EQL 0                                            ! And level agrees that really no mo
; 1289        2007  5                      THEN prompt_flags = .prompt_flags OR hcf$m_stay;        ! Then set prompt flag
; 1290        2008  5                   truekeys [0] = .level;                                     ! Set number of keys
; 1291        2009  5                   EXITLOOP;                                                  ! And exit
; 1292        2010  4                   END;
; 1293        2011  4           curkeydesc [dsc$a_pointer] = .parm_begin;                         !Set pointer to start of key
; 1294        2012  3           END;
; 1295        2013  2       END;                                                                  !Of INCRU loop
; 1296        2014  2
; 1297        2015  2 RETURN true
; 1298        2016  2
; 1299        2017  1 END;                                                                        !Of setup_keys



                                                   00FC 00000 SETUP_KEYS:
                                                                  .WORD   Save R2,R3,R4,R5,R6,R7                    ; 1914
                                     57           28  D0 00002    MOVL    #40, PAREN                                ; 1952
   0050   8F           00            6E           00  2C 00005    MOVC5   #0, (SP), #0, #80, @KEYDESCS              ; 1953
```

```
                                     08    BC    0000C                                                 :
                         53          04    AC D0 0000E             MOVL    GETCMDDESC, R3              : 1960
                                     04    A3 DD 00012             PUSHL   4(R3)                       :
                                           53 DD 00015             PUSHL   R3                          :
                  0000V  CF                02 FB 00017             CALLS   #2, MAKE_UPPER_CASE         :
                         55          04    A3 D0 0001C             MOVL    4(R3), PARM_END             : 1961
                         52                63 3C 00020             MOVZWL  (R3), LINE_END              : 1962
                         52          04    A3 C0 00023             ADDL2   4(R3), LINE_END             :
      04    B3           63                2F 3A 00027             LOCC    #47, (R3), @4(R3)           : 1963
                                           02 12 0002C             BNEQ    1$                          :
                                           51 D4 0002E             CLRL    R1                          :
                         56                51 D0 00030 1$:         MOVL    R1, NEXT_QUAL               :
                                           54 D4 00033             CLRL    LEVEL                       : 1970
            50           52                55 C3 00035 2$:         SUBL3   PARM_END, LINE_END, R0      :
            65           50                20 3B 00039             SKPC    #32, R0, (PARM_END)         :
                                           02 12 0003D             BNEQ    3$                          :
                                           51 D4 0003F             CLRL    R1                          :
                         53                51 D0 00041 3$:         MOVL    R1, PARM_BEGIN              :
                                           03 12 00044             BNEQ    4$                          : 1971
                         53                52 D0 00046             MOVL    LINE_END, PARM_BEGIN        : 1972
                                           56 D5 00049 4$:         TSTL    NEXT_QUAL                   : 1974
                                           1A 13 0004B             BEQL    6$                          :
                         56                53 D1 0004D             CMPL    PARM_BEGIN, NEXT_QUAL       : 1975
                                           15 1F 00050             BLSSU   6$                          :
                         53                56 D0 00052             MOVL    NEXT_QUAL, PARM_BEGIN       : 1977
            50           52                53 C3 00055             SUBL3   PARM_BEGIN, LINE_END, R0    : 1978
                                           50 D7 00059             DECL    R0                          :
      01    A3           50                2F 3A 0005B             LOCC    #47, R0, 1(PARM_BEGIN)      :
                                           02 12 00060             BNEQ    5$                          :
                                           51 D4 00062             CLRL    R1                          :
                         56                51 D0 00064 5$:         MOVL    R1, NEXT_QUAL               :
            50           52                53 C3 00067 6$:         SUBL3   PARM_BEGIN, LINE_END, R0    : 1982
            63           50                20 3A 0006B             LOCC    #32, R0, (PARM_BEGIN)       :
                                           02 12 0006F             BNEQ    7$                          :
                                           51 D4 00071             CLRL    R1                          :
                         55                51 D0 00073 7$:         MOVL    R1, PARM_END                :
                                           03 12 00076             BNEQ    8$                          : 1983
                         55                52 D0 00078             MOVL    LINE_END, PARM_END          : 1984
                                           56 D5 0007B 8$:         TSTL    NEXT_QUAL                   : 1986
                                           08 13 0007D             BEQL    9$                          :
                         56                55 D1 0007F             CMPL    PARM_END, NEXT_QUAL         : 1987
                                           03 1B 00082             BLEQU   9$                          :
                         55                56 D0 00084             MOVL    NEXT_QUAL, PARM_END         : 1988
                         53                55 D1 00087 9$:         CMPL    PARM_END, PARM_BEGIN        : 1992
                                           05 13 0008A             BEQL    10$                         :
                         57                63 91 0008C             CMPB    (PARM_BEGIN), PAREN         : 1993
                                           A4 13 0008F             BEQL    2$                          :
                         50       08 BC44  7E 00091 10$:        MOVAQ   @KEYDESCS[LEVEL], R0        : 2002
            51           55                53 C3 00096             SUBL3   PARM_BEGIN, PARM_END, R1    : 2004
                         60                51 B0 0009A             MOVW    R1, (R0)                    :
                                           51 D5 0009D             TSTL    R1                          :
                                           0F 12 0009F             BNEQ    12$                         :
                                           54 D5 000A1             TSTL    LEVEL                       : 2006
                                           05 12 000A3             BNEQ    11$                         :
                  0000'  CF                02 88 000A5             BISB2   #2, PROMPT_FLAGS            : 2007
                     0C  BC                54 90 000AA 11$:        MOVB    LEVEL, @TRUEKEYS            : 2008
                                           0E 11 000AE             BRB     13$                         : 2005
```

```
                              04    A0              53  D0 000B0 12$:     MOVL    PARM_BEGIN, 4(R0)                       ; 2011
                                                    54  D6 000B4          INCL    LEVEL                                   ; 1966
                                    09              54  D1 000B6          CMPL    LEVEL, #9                               ;
                                                    03  1A 000B9          BGTRU   13$                                     ;
                                                  FF77  31 000BB          BRW     2$                                      ;
                                    50              01  D0 000BE 13$:     MOVL    #1, R0                                  ; 2015
                                                        04 000C1          RET                                             ; 2017

; Routine Size:  194 bytes,    Routine Base:  $CODE$ + 09A0
```

```
1 %SBTTL 'Routine call_lbrhelp';
1 ROUTINE call_lbrhelp (helplibindex, outputwidth, printdata, keydescs) =
2 BEGIN
2
2 !++
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       This routine calls the librarian function to extract help from
2 !       a particular help library.
2 !
2 ! INPUTS:
2 !
2 !       helplibindex =  address of longword containing help library index
2 !
2 !       outputwidth =   address of longword containing width of output line
2 !
2 !       printdata =     address of data structure containg information for
2 !                       the output driver
2 !
2 !       keydescs =      address of vector of individual descriptors for each keyword
2 !
2 ! OUTPUTS:
2 !
2 !       None.
2 !
2 ! ROUTINE VALUE:
2 !
2 !       Status of call for help.
2 !
2 !--
2
2 MAP
2     keydescs : REF VECTOR [,BYTE];
2
2 LOCAL
2     status:     BBLOCK [LONG];
2
2 EXTERNAL
2     lbr$gl_control : REF BBLOCK;                 ! Pointer to current library control block
2
2 BIND
2     context = .lbr$gl_control [lbr$l_ctxptr] : BBLOCK;
2
2 !
2 ! Set this bit to make LBR$GET_HELP handle the help on help case correctly.
2 ! This bit is not universally set for V2.0 compatibility reasons.
2 !
2 context [ctx$v_outputhlp] = true;
2 status = lbr$get_help (.helplibindex, .outputwidth,
2                 output_driver, .printdata,
2                 keydescs [dsc$c_s_bln * 0], keydescs [dsc$c_s_bln * 1],
2                 keydescs [dsc$c_s_bln * 2], keydescs [dsc$c_s_bln * 3],
2                 keydescs [dsc$c_s_bln * 4], keydescs [dsc$c_s_bln * 5],
2                 keydescs [dsc$c_s_bln * 6], keydescs [dsc$c_s_bln * 7],
2                 keydescs [dsc$c_s_bln * 8], keydescs [dsc$c_s_bln * 9]);
2 context [ctx$v_outputhlp] = false;
2
```

```
; 1358          2075  2 IF .status EQL lbr$_endtopic THEN                   ! Abort help on this topic?
; 1359          2076  3     BEGIN
; 1360          2077  3     end_topic_flag = true;                          ! Set flag to abort additional library listing and
; 1361          2078  3     status = true;                                  ! change error to true
; 1362          2079  2     END;
; 1363          2080  2
; 1364          2081  2 RETURN .status
; 1365          2082  1 END;                                                ! Of call_lbrhelp


                                                                    .EXTRN  LBR$GL_CONTROL

                                              0004 00000 CALL_LBRHELP:
                                                                    .WORD   Save R2                                 ; 2019
                          50      0000G   CF   D0 00002            MOVL    LBR$GL_CONTROL, R0                      ; 2059
                          52         0E   A0   D0 00007            MOVL    14(R0), R2                              ;
                     05   A2              01   88 0000B            BISB2   #1, 5(R2)                               ; 2065
                          50         10   AC   D0 0000F            MOVL    KEYDESCS, R0                            ; 2072
                                     48   A0   9F 00013            PUSHAB  72(R0)                                  ;
                                     40   A0   9F 00016            PUSHAB  64(R0)                                  ;
                                     38   A0   9F 00019            PUSHAB  56(R0)                                  ; 2071
                                     30   A0   9F 0001C            PUSHAB  48(R0)                                  ;
                                     28   A0   9F 0001F            PUSHAB  40(R0)                                  ; 2070
                                     20   A0   9F 00022            PUSHAB  32(R0)                                  ;
                                     18   A0   9F 00025            PUSHAB  24(R0)                                  ; 2069
                                     10   A0   9F 00028            PUSHAB  16(R0)                                  ;
                                     08   A0   9F 0002B            PUSHAB  8(R0)                                   ; 2068
                                          50   DD 0002E            PUSHL   R0                                      ; 2072
                                     0C   AC   DD 00030            PUSHL   PRINTDATA                               ;
                                  0000V   CF   9F 00033            PUSHAB  OUTPUT_DRIVER                           ; 2066
                          7E         04   AC   7D 00037            MOVQ    HELPLIBINDEX, -(SP)                     ; 2072
              00000000G   00              0E   FB 0003B            CALLS   #14, LBR$GET_HELP                       ;
                     05   A2              01   8A 00042            BICB2   #1, 5(R2)                               ; 2073
              00000000G   8F              50   D1 00046            CMPL    STATUS, #LBR$_ENDTOPIC                  ; 2075
                                          08   12 0004D            BNEQ    1$                                      ;
                  0000'   CF              01   90 0004F            MOVB    #1, END_TOPIC_FLAG                      ; 2077
                          50              01   D0 00054            MOVL    #1, STATUS                              ; 2078
                                               04 00057 1$:        RET                                             ; 2082

; Routine Size:  88 bytes,    Routine Base:  $CODE$ + 0A62
```

```
%SBTTL 'Routine output_driver';
ROUTINE output_driver (linedesc, helpflags, printdata, helplevel) =
BEGIN

!++
! FUNCTIONAL DESCRIPTION:
!
!       Call user supplied output routine to print a line of help text
!       Also set various prompt flags and set initialize pointer user for
!       putting keys into subprompt buffer.
!
! INPUTS:
!
!       linedesc =      address of descriptor for line to be output
!
!       helpflags =     address of flag longword describing contents of
!                       text that is passed
!
!       printdata =     address of data structure containing flags, levels,
!                       and other data for the output driver
!
!       helplevel =     address of longword containing the current key level
!
! OUTPUTS:
!
!       The various data items in printdata are updated and the user supplied
!       output routine is called to output the linedesc.
!
! ROUTINE VALUE:
!
!       False, if call to user supplied routine returns false.
!       True, otherwise.
!--

MAP
    linedesc : REF BBLOCK,
    printdata : REF BBLOCK;

BIND
    true_keys = printdata [hpd$b_truekeys] : SIGNED BYTE,       ! Number of help keys
    help_level = printdata [hpd$b_helplevel] : BYTE,            ! Current key depth
    print_flags = printdata [hpd$b_printflag] : BBLOCK,         ! Flags for output driver
    add_info_level = printdata [hpd$l_subpmtlev],               ! Current prompt level
    sub_prompt_ptr = printdata [hpd$l_subpmtptr],               ! Ptr used for filling sub-prompt buffer
    length_array = printdata [hpd$l_lenarray] : REF VECTOR,     ! Address of key length array
    output_routine = printdata [hpd$l_outputrou];               ! User specified output routine

OWN
    topics_available : COUNTEDSTRING ('Information available:');

LOCAL
    keylevel,                                                   ! Local equivalent of parameter helplevel
    flags,                                                      ! Local equivelent of parameter helpflags
    output_buf : BBLOCK [hlp$c_pagesize],                       ! Local buffer to store line to be output
    output_desc : BBLOCK [dsc$c_s_bln],                         ! Local output descriptor
    ptr,                                                        ! Pointer into output buffer
    spaces : WORD,                                              ! Number of spaces to indent output line
```

```
2           status;
2
2       status = true;                                                  ! Status if user output routine never called
2       flags = ..helpflags;                                            ! Copy helpflags locally
2
2       IF NOT .print_flags [hpd$v_all] AND                             ! If not printing all help text
3          ((.flags AND hlp$m_nohlptxt) NEQ 0)                          ! and no help text found
3          THEN BEGIN
3               print_flags [hpd$v_found] = false;                      ! Then set no help found flag
3               RETURN true;                                            ! And return
2               END;
2
2       print_flags [hpd$v_found] = true;                               ! Else say help was found
2       print_flags [hpd$v_all] = true;                                 ! Keyword was found, so this will ensure tha
2                                                                       !  will not be search, even if no text follo
2       keylevel = ..helplevel;                                         ! Copy key level locally
2       output_desc = 0;                                                ! Init local descriptor for output line
2       output_desc [dsc$w_length] = .linedesc [dsc$w_length];
2       output_desc [dsc$a_pointer] = output_buf;
2
2       IF .flags EQL 0                                                 ! If help info found
2          THEN prompt_flags = .prompt_flags OR hcf$m_info;             ! Then say so
2
2       !
2       ! If processing key name and moving down a prompt level,
2       ! then build subtopic prompt
2       !
2
2       IF ((.flags AND hlp$m_keynamlin) NEQ 0) AND
3          ((.prompt_flags AND hcf$m_stay) EQL 0)
3          THEN BEGIN
3               IF (.prompt_flags AND hcf$m_info) NEQ 0
3                  THEN prompt_flags = .prompt_flags OR hcf$m_stay
4                  ELSE BEGIN
5                       IF (.keylevel NEQ .help_level) AND (.output_desc [dsc$w_length] NEQ 0)
5                          THEN BEGIN
5                               help_level = .keylevel;
5                               CH$MOVE (.output_desc [dsc$w_length], .linedesc [dsc$a_pointer],
5                                         .sub_prompt_ptr);
5                               length_array [.help_level] = .output_desc [dsc$w_length];
5                               sub_prompt_ptr = .sub_prompt_ptr + .output_desc [dsc$w_length];
5                               (.sub_prompt_ptr)<0,8> = 32;
5                               sub_prompt_ptr = .sub_prompt_ptr + 1;
4                               END;
3                       END;
3               add_info_level = .keylevel;
2               END;
2
2       !
2       ! If no help text found, then stay at current level and signal that
2       ! help was not found.
2       !
2
2       IF (.flags AND hlp$m_nohlptxt) NEQ 0
3               THEN BEGIN
3                    prompt_flags = .prompt_flags OR hcf$m_stay;
4                    IF (.print_flags [hpd$v_init] EQL 0)
```

```
3                       THEN print_flags [hpd$v_init] = 1;
2                   END;
2
2 !
2 ! If additional info to be found, say so
2 !
2
2 IF (.flags AND hlp$m_otherinfo) NEQ 0
2    THEN prompt_flags = .prompt_flags OR hcf$m_more;
2
2 !
2 ! Format output line and then call user supplied output routine.
2 !
2
3 IF ((.true_keys NEQ 0) OR                             ! Not at first topic level
3    .help_flags [hlp$v_help] OR                        ! Want help on help with topic list
3    ((.flags AND hlp$m_otherinfo) NEQ 0))              ! List of additional info
2    THEN IF (.keylevel GTR 0) OR
3            ((.flags AND hlp$m_keynamlin) EQL 0)
3            THEN BEGIN
3                 IF .linedesc [dsc$w_length] NEQ 0
4                    THEN BEGIN
4
5                         IF ((.flags AND hlp$m_keynamlin) NEQ 0)
4                            THEN spaces = (.keylevel - 1) * hlp$c_indent
5                            ELSE IF ((.flags AND hlp$m_nohlptxt) NEQ 0)
5                                    AND (.keylevel EQL 0)
4                                    THEN spaces = hlp$c_indent
4                                    ELSE spaces = .keylevel * hlp$c_indent;
4
4                         ptr = CH$FILL (%ASCII ' ' , .spaces, output_buf);
5                         IF ((.true_keys EQL 0) AND NOT .help_flags [hlp$v_help])
5                            THEN BEGIN
5                                 true_keys = -1;
5                                 output_desc [dsc$w_length] = .topics_available [0];
5                                 CH$MOVE (.output_desc [dsc$w_length], topics_available + 1, .ptr);
5                                 END
5                            ELSE BEGIN
5                                 CH$MOVE (.output_desc [dsc$w_length], .linedesc [dsc$a_pointer], .ptr);
4                                 END;
4                         output_desc [dsc$w_length] = .spaces + .output_desc [dsc$w_length];
3                         END;
3
3                 status = (.output_routine) (output_desc);
3
2                 END;
2
2 RETURN .status;
1 END;                                                  !Of output_driver


                                                         .PSECT  $OWN$,NOEXE,2

                                              16  00054 TOPICS_AVAILABLE:
                                                         .BYTE   22
61 76 61 20 6E 6F 69 74 61 6D 72 6F 66 6E 49  00055     .ASCII  \Information available:\                     ;
```

```
                          3A  65  6C  62  61  6C  69  00064                                                                 ;


                                                                      .PSECT   $CODE$,NOWRT,2

                                              0FFC 00000 OUTPUT_DRIVER:
                                                                      .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11         ; 2084
                              5E     FDF8  CE   9E 00002              MOVAB    -520(SP), SP                                 ;
                              57       0C  AC   D0 00007              MOVL     PRINTDATA, R7                                ; 2122
                              59       12  A7   9E 0000B              MOVAB    18(R7), R9                                   ; 2124
                                           01   DD 0000F              PUSHL    #1                                           ; 2142
                              5A       08  BC   D0 00011              MOVL     @HELPFLAGS, FLAGS                            ; 2143
                  0A          69           01   E0 00015              BBS      #1, (R9), 1$                                 ; 2145
                              07           5A   E9 00019              BLBC     FLAGS, 1$                                    ; 2146
                              69           04   8A 0001C              BICB2    #4, (R9)                                     ; 2148
                              50           01   D0 0001F              MOVL     #1, R0                                       ; 2149
                                                04 00022              RET                                                   ;
                              69           06   88 00023 1$:          BISB2    #6, (R9)                                     ; 2153
                              56       10  BC   D0 00026              MOVL     @HELPLEVEL, KEYLEVEL                         ; 2155
                                       04  AE   D4 0002A              CLRL     OUTPUT_DESC                                  ; 2156
                              58       04  AC   D0 0002D              MOVL     LINEDESC, R8                                 ; 2157
                         04   AE           68   B0 00031              MOVW     (R8), OUTPUT_DESC                            ;
                         08   AE       0C  AE   9E 00035              MOVAB    OUTPUT_BUF, OUTPUT_DESC+4                    ; 2158
                                           5A   D5 0003A              TSTL     FLAGS                                        ; 2160
                                           05   12 0003C              BNEQ     2$                                           ;
                      0000'   CF           08   88 0003E              BISB2    #8, PROMPT_FLAGS                             ; 2161
                                           5B   D4 00043 2$:          CLRL     R11                                          ; 2168
                  48          5A           01   E1 00045              BBC      #1, FLAGS, 5$                                ;
                                           5B   D6 00049              INCL     R11                                          ;
                  40  0000'   CF           01   E0 0004B              BBS      #1, PROMPT_FLAGS, 5$                         ; 2169
                  07  0000'   CF           03   E1 00051              BBC      #3, PROMPT_FLAGS, 3$                         ; 2171
                      0000'   CF           02   88 00057              BISB2    #2, PROMPT_FLAGS                             ; 2172
                                           2F   11 0005C              BRB      4$                                           ;
     56       11  A7              08       00   ED 0005E 3$:          CMPZV    #0, #8, 17(R7), KEYLEVEL                     ; 2174
                                           27   13 00064              BEQL     4$                                           ;
                                       04  AE   B5 00066              TSTW     OUTPUT_DESC                                  ;
                                           22   13 00069              BEQL     4$                                           ;
                         11   A7           56   90 0006B              MOVB     KEYLEVEL, 17(R7)                             ; 2176
              00  B7     04   B8       04  AE   28 0006F              MOVC3    OUTPUT_DESC, @4(R8), @0(R7)                  ; 2178
                              50       11  A7   9A 00076              MOVZBL   17(R7), R0                                   ; 2179
                         0C B740       04  AE   3C 0007A              MOVZWL   OUTPUT_DESC, @12(R7)[R0]                     ;
                              50       04  AE   3C 00080              MOVZWL   OUTPUT_DESC, R0                              ; 2180
                              67           50   C0 00084              ADDL2    R0, (R7)                                     ;
                         00   B7           20   90 00087              MOVB     #32, @0(R7)                                  ; 2181
                                           67   D6 0008B              INCL     (R7)                                         ; 2182
                         04   A7           56   D0 0008D 4$:          MOVL     KEYLEVEL, 4(R7)                              ; 2185
                                           51   D4 00091 5$:          CLRL     R1                                           ; 2193
                              0D           5A   E9 00093              BLBC     FLAGS, 6$                                    ;
                                           51   D6 00096              INCL     R1                                           ;
                      0000'   CF           02   88 00098              BISB2    #2, PROMPT_FLAGS                             ; 2195
                              03           69   E8 0009D              BLBS     (R9), 6$                                     ; 2196
                              69           01   88 000A0              BISB2    #1, (R9)                                     ; 2197
                                           50   D4 000A3 6$:          CLRL     R0                                           ; 2204
                  07          5A           02   E1 000A5              BBC      #2, FLAGS, 7$                                ;
                                           50   D6 000A9              INCL     R0                                           ;
                      0000'   CF           04   88 000AB              BISB2    #4, PROMPT_FLAGS                             ; 2205
```

```
                                        10    A7  95 000B0 7$:      TSTB      16(R7)                                      ; 2211
                                              09  12 000B3          BNEQ      8$                                          ;
                   03     0000'  CF           05  E0 000B5          BBS       #5, HELP_FLAGS, 8$                          ; 2212
                                 62           50  E9 000BB          BLBC      R0, 16$                                     ; 2213
                                              56  D5 000BE 8$:      TSTL      KEYLEVEL                                    ; 2214
                                              04  14 000C0          BGTR      9$                                          ;
                   5A            5A           01  E0 000C2          BBS       #1, FLAGS, 16$                              ; 2215
                                              68  B5 000C6 9$:      TSTW      (R8)                                        ; 2217
                                              4C  13 000C8          BEQL      15$                                         ;
                                 0A           5B  E9 000CA          BLBC      R11, 10$                                    ; 2220
                                 50     FF    A6  9E 000CD          MOVAB     -1(R6), R0                                  ; 2221
                   5A            50           02  A5 000D1          MULW3     #2, R0, SPACES                              ;
                                              10  11 000D5          BRB       12$                                         ;
                                 09           51  E9 000D7 10$:     BLBC      R1, 11$                                     ; 2222
                                              56  D5 000DA          TSTL      KEYLEVEL                                    ; 2223
                                              05  12 000DC          BNEQ      11$                                         ;
                                 5A           02  B0 000DE          MOVW      #2, SPACES                                  ; 2224
                                              04  11 000E1          BRB       12$                                         ;
                   5A            56           02  A5 000E3 11$:     MULW3     #2, KEYLEVEL, SPACES                        ; 2225
        5A         20            6E           00  2C 000E7 12$:     MOVC5     #0, (SP), #32, SPACES, OUTPUT_BUF           ; 2227
                                        0C    AE     000EC                                                                ;
                                        10    A7  95 000EE          TSTB      16(R7)                                      ; 2228
                                              19  12 000F1          BNEQ      13$                                         ;
                   13     0000'  CF           05  E0 000F3          BBS       #5, HELP_FLAGS, 13$                         ;
                            10   A7           01  8E 000F9          MNEGB     #1, 16(R7)                                  ; 2230
                            04   AE   0000'   CF  9B 000FD          MOVZBW    TOPICS_AVAILABLE, OUTPUT_DESC               ; 2231
                   63     0000'  CF     04    AE  28 00103          MOVC3     OUTPUT_DESC, TOPICS_AVAILABLE+1, (PTR)      ; 2232
                                              06  11 0010A          BRB       14$                                         ; 2228
                   63       04   B8     04    AE  28 0010C 13$:     MOVC3     OUTPUT_DESC, @4(R8), (PTR)                  ; 2235
                            04   AE           5A  A0 00112 14$:     ADDW2     SPACES, OUTPUT_DESC                         ; 2237
                                        04    AE  9F 00116 15$:     PUSHAB    OUTPUT_DESC                                 ; 2240
                            08   B7           01  FB 00119          CALLS     #1, @8(R7)                                  ;
                                 6E           50  D0 0011D          MOVL      R0, STATUS                                  ;
                                 50           6E  D0 00120 16$:     MOVL      STATUS, R0                                  ; 2244
                                                  04 00123          RET                                                   ; 2245

; Routine Size:  292 bytes,    Routine Base:  $CODE$ + 0ABA
```

```
1 %SBTTL 'Routine libs_available';
1 ROUTINE libs_available (outputroutine, outputwidth) =
2 BEGIN
2
2 !++
2 ! FUNCTIONAL DESCRIPTION
2 !
2 !       Output a list of the define default libraries.
2 !
2 ! INPUTS:
2 !
2 !       outputroutine = address of the user supplied output routine
2 !
2 !       outputwidth =   address of a longword containing the output line width
2 !
2 ! OUTPUTS:
2 !
2 !       The list is output by using the user supplied routine.
2 !
2 ! ROUTINE VALUE:
2 !
2 !       Always true.
2 !--
2
2 LOCAL
2     acmode,
2     blank_line : BBLOCK [dsc$c_s_bln],
2     filespec : BBLOCK [dsc$c_s_bln],
2     filespec_buf : BBLOCK [nam$c_maxrss],
2     filename : BBLOCK [dsc$c_s_bln],
2     filename_buf : VECTOR [filename_length, BYTE],
2     libno,
2     output_desc : BBLOCK [dsc$c_s_bln],
2     output_buf : BBLOCK [hlp$c_pagesize],
2     status;
2
2 OWN
2     header : COUNTEDSTRING ('  Additional help libraries available (type @name for topics):');
2
2 filespec = 0;                                         ! Init file spec desc
2 filespec [dsc$a_pointer] = filespec_buf;
2
2 libno = -1;                                           ! Init search
2 status = false;                                       ! Init status
2
2 WHILE NOT .status                                     ! Loop searching for a library
3 DO BEGIN
3    IF NOT tran_next_lib (filespec, acmode, libno)     ! Get first lib spec
3       THEN RETURN true;                               ! If none, then return
3    status = file_present (filespec);                  ! Is file present?
2    END;
2
2 filename = 0;                                         ! Init file name desc
2 filename [dsc$a_pointer] = filename_buf;
2
2 blank_line = 0;                                       ! Init blank line desc
2 blank_line [dsc$a_pointer] = output_buf;
```

```
; 1588          2303  2
; 1589          2304  2 output_desc = .header [0];                          ! Init header desc
; 1590          2305  2 output_desc [dsc$a_pointer] = header [1];
; 1591          2306  2
; 1592          2307  2 (.outputroutine) (blank_line);                       ! Output header
; 1593          2308  2 (.outputroutine) (output_desc);
; 1594          2309  2 (.outputroutine) (blank_line);
; 1595          2310  2
; 1596          2311  2 output_desc = 0;                                     ! Reuser output desc for
; 1597          2312  2 output_desc [dsc$a_pointer] = output_buf;            ! lib name lists
; 1598          2313  2 CH$FILL (%X'20', ..outputwidth, output_buf);
; 1599          2314  2
; 1600          2315  2 switch_libname (filespec, filename);                  ! Get first lib name
; 1601          2316  2 output_desc [dsc$w_length] = .filename[dsc$w_length] + 2;       ! Move it into
; 1602          2317  2 CH$MOVE (.filename [dsc$w_length], filename_buf, output_buf + 2); ! the list.
; 1603          2318  2
; 1604          2319  2 WHILE tran_next_lib(filespec, acmode, libno) DO       ! While more libs translate,
; 1605          2320  2     IF file_present(filespec) THEN                   ! if the file exists,
; 1606          2321  3         BEGIN
; 1607          2322  3         switch_libname(filespec, filename);          ! get its name.
; 1608          2323  3
; 1609          2324  3         ! If we can't fit a couple of spaces and the filename on the
; 1610          2325  3         ! line we're working on, send the line to the output routine and
; 1611          2326  3         ! start a new one:
; 1612          2327  3         !
; 1613          2328  3         IF (.output_desc[dsc$w_length] + 2 + .filename[dsc$w_length]) GTRU ..outputwidth THEN
; 1614          2329  4             BEGIN
; 1615          2330  4             (.outputroutine) (output_desc);
; 1616          2331  4             output_desc[dsc$w_length] = 0;
; 1617          2332  4             CH$FILL (%X'20', ..outputwidth, output_buf);
; 1618          2333  3             END;
; 1619          2334  3
; 1620          2335  3         ! Move the filename onto the line (leaving two padding spaces):
; 1621          2336  3         !
; 1622          2337  3         CH$MOVE (.filename[dsc$w_length], filename_buf,
; 1623          2338  3                     output_buf + .output_desc[dsc$w_length] + 2);
; 1624          2339  3         output_desc[dsc$w_length] = .output_desc[dsc$w_length] + 2 + .filename[dsc$w_length];
; 1625          2340  2         END;
; 1626          2341  2
; 1627          2342  2 (.outputroutine) (output_desc);                      ! Output last line built
; 1628          2343  2 RETURN true.
; 1629          2344  1 END;


                                                                 .PSECT  $OWN$,NOEXE,2

                                                  0006B          .BLKB   1
                                              3E  0006C HEADER:  .BYTE   62
65 68 20 6C 61 6E 6F 69 74 69 64 64 41 20 20  0006D          .ASCII  \  Additional help libraries available (t\    ;
76 61 20 73 65 69 72 61 72 62 69 6C 20 70 6C  0007C                                                             ;
               74 28 20 65 6C 62 61 6C 69 61  0008B                                                             ;
74 20 72 6F 66 20 65 6D 61 6E 40 20 65 70 79  00095          .ASCII  \ype @name for topics):\                     ;
                        3A 29 73 63 69 70 6F  000A4                                                             ;
```

```
                                                                   .PSECT  $CODE$,NOWRT,2

                                            03FC 00000 LIBS_AVAILABLE:
                                                                   .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9                       ; 2247
                              59      FC2B    CF  9E 00002         MOVAB   TRAN_NEXT_LIB, R9                                  ;
                              5E      FCB4    CE  9E 00007         MOVAB   -844(SP), SP                                       ;
                                        F0    AD  D4 0000C         CLRL    FILESPEC                                           ; 2285
                        F4    AD      FEF0    CD  9E 0000F         MOVAB   FILESPEC_BUF, FILESPEC+4                           ; 2286
                              7E              01  CE 00015         MNEGL   #1, LIBNO                                          ; 2288
                                              52  D4 00018         CLRL    STATUS                                             ; 2289
                              1E              52  E8 0001A 1$:     BLBS    STATUS, 3$                                         ; 2291
                                              5E  DD 0001D         PUSHL   SP                                                 ; 2293
                                        08    AE  9F 0001F         PUSHAB  ACMODE                                             ;
                                        F0    AD  9F 00022         PUSHAB  FILESPEC                                           ;
                              69              03  FB 00025         CALLS   #3, TRAN_NEXT_LIB                                  ;
                              03              50  E8 00028         BLBS    R0, 2$                                             ;
                                          00DF    31 0002B         BRW     7$                                                 ;
                                        F0    AD  9F 0002E 2$:     PUSHAB  FILESPEC                                           ; 2295
                    0000V     CF              01  FB 00031         CALLS   #1, FILE_PRESENT                                   ;
                              52              50  D0 00036         MOVL    R0, STATUS                                         ;
                                              DF  11 00039         BRB     1$                                                 ; 2291
                                      FEE8    CD  D4 0003B 3$:     CLRL    FILENAME                                           ; 2298
                    FEEC      CD      FEC0    CD  9E 0003F         MOVAB   FILENAME_BUF, FILENAME+4                           ; 2299
                                        F8    AD  D4 00046         CLRL    BLANK_LINE                                         ; 2301
                      FC      AD        08    AE  9E 00049         MOVAB   OUTPUT_BUF, BLANK_LINE+4                           ; 2302
                    FEB8      CD      0000'   CF  9A 0004E         MOVZBL  HEADER, OUTPUT_DESC                                ; 2304
                    FEBC      CD      0000'   CF  9E 00055         MOVAB   HEADER+1, OUTPUT_DESC+4                            ; 2305
                              58        04    AC  D0 0005C         MOVL    OUTPUTROUTINE, R8                                  ; 2307
                                        F8    AD  9F 00060         PUSHAB  BLANK_LINE                                         ;
                              68              01  FB 00063         CALLS   #1, (R8)                                           ;
                                      FEB8    CD  9F 00066         PUSHAB  OUTPUT_DESC                                        ; 2308
                              68              01  FB 0006A         CALLS   #1, (R8)                                           ;
                                        F8    AD  9F 0006D         PUSHAB  BLANK_LINE                                         ; 2309
                              68              01  FB 00070         CALLS   #1, (R8)                                           ;
                                      FEB8    CD  D4 00073         CLRL    OUTPUT_DESC                                        ; 2311
                    FEBC      CD        08    AE  9E 00077         MOVAB   OUTPUT_BUF, OUTPUT_DESC+4                          ; 2312
     08    BC          20     6E              00  2C 0007D         MOVC5   #0, (SP), #32, @OUTPUTWIDTH, OUTPUT_BUF            ; 2313
                                        08    AE     00083                                                                    ;
                                      FEE8    CD  9F 00085         PUSHAB  FILENAME                                           ; 2315
                                        F0    AD  9F 00089         PUSHAB  FILESPEC                                           ;
                      85      A9              02  FB 0008C         CALLS   #2, SWITCH_LIBNAME                                 ;
       FEB8    CD   FEE8      CD              02  A1 00090         ADDW3   #2, FILENAME, OUTPUT_DESC                          ; 2316
         0A    AE   FEC0      CD      FEE8    CD  28 00098         MOVC3   FILENAME, FILENAME_BUF, OUTPUT_BUF+2               ; 2317
                                              5E  DD 000A1 4$:     PUSHL   SP                                                 ; 2319
                                        08    AE  9F 000A3         PUSHAB  ACMODE                                             ;
                                        F0    AD  9F 000A6         PUSHAB  FILESPEC                                           ;
                              69              03  FB 000A9         CALLS   #3, TRAN_NEXT_LIB                                  ;
                              57              50  E9 000AC         BLBC    R0, 6$                                             ;
                                        F0    AD  9F 000AF         PUSHAB  FILESPEC                                           ; 2320
                    0000V     CF              01  FB 000B2         CALLS   #1, FILE_PRESFNT                                   ;
                              E7              50  E9 000B7         BLBC    R0, 4$                                             ;
                                      FEE8    CD  9F 000BA         PUSHAB  FILENAME                                           ; 2322
                                        F0    AD  9F 000BE         PUSHAB  FILESPEC                                           ;
                      85      A9              02  FB 000C1         CALLS   #2, SWITCH_LIBNAME                                 ;
                              50      FEB8    CD  3C 000C5         MOVZWL  OUTPUT_DESC, R0                                    ; 2328
                              57      FEE8    CD  3C 000CA         MOVZWL  FILENAME, R7                                       ;
                              50        02  A740  9E 000CF         MOVAB   2(R7)[R0], R0                                      ;
                              08   BC              50  D1 000D4         CMPL    R0, @OUTPUTWIDTH                        ;
                                                   13  1B 000D8         BLEQU   5$                                      ;
                                            FEB8   CD  9F 000DA         PUSHAB  OUTPUT_DESC                             ; 2330
                                    68             01  FB 000DE         CALLS   #1, (R8)                                ;
                                            FEB8   CD  B4 000E1         CLRW    OUTPUT_DESC                             ; 2331
     08   BC          20            6E             00  2C 000E5         MOVC5   #0, (SP), #32, @OUTPUTWIDTH, OUTPUT_BUF ; 2332
                                              08   AE     000EB                                                         ;
                                    56      FEB8   CD  3C 000ED 5$:     MOVZWL  OUTPUT_DESC, R6                         ; 2338
                    0A AE46    FEC0 CD             57  28 000F2         MOVC3   R7, FILENAME_BUF, OUTPUT_BUF+2[R6]      ;
                                    50         02 A746 9E 000FA         MOVAB   2(R7)[R6], R0                           ; 2339
                               FEB8 CD             50  B0 000FF         MOVW    R0, OUTPUT_DESC                         ;
                                                   9B  11 00104         BRB     4$                                      ; 2320
                                            FEB8   CD  9F 00106 6$:     PUSHAB  OUTPUT_DESC                             ; 2342
                                    68             01  FB 0010A         CALLS   #1, (R8)                                ;
                                    50             01  D0 0010D 7$:     MOVL    #1, R0                                  ; 2343
                                                       04 00110         RET                                             ; 2344

; Routine Size:  273 bytes,    Routine Base:  $CODE$ + 0BDE
; 1631             2345  1 %SBTTL 'Routine file_present';
; 1632             2346  1 ROUTINE file_present (filename) =
; 1633             2347  2 BEGIN
; 1634             2348  2
; 1635             2349  2 !++
; 1636             2350  2 ! FUNCTIONAL DESCRIPTION:
; 1637             2351  2 !
; 1638             2352  2 !       Return success if the file exists.
; 1639             2353  2 !
; 1640             2354  2 ! INPUTS:
; 1641             2355  2 !
; 1642             2356  2 !       filename = address of desc of file name
; 1643             2357  2 !
; 1644             2358  2 ! OUTPUTS:
; 1645             2359  2 !
; 1646             2360  2 !       None.
; 1647             2361  2 !
; 1648             2362  2 ! ROUTINE VALUE:
; 1649             2363  2 !
; 1650             2364  2 !       True, if file exists
; 1651             2365  2 !       False, if it doesn't
; 1652             2366  2 !
; 1653             2367  2 !--
; 1654             2368  2
; 1655             2369  2 MAP
; 1656             2370  2     filename : REF BBLOCK;
; 1657             2371  2
; 1658             2372  2 LOCAL
; 1659             2373  2     fab : BBLOCK [fab$c_bln],
; 1660             2374  2     nam : BBLOCK [nam$c_bln],
; 1661             2375  2     string : BBLOCK [nam$c_maxrss],
; 1662             2376  2     status;
; 1663             2377  2
; 1664           P 2378  2 $NAM_INIT ( NAM = nam,
; 1665           P 2379  2             ESS = nam$c_maxrss,
; 1666             2380  2             ESA = string);
; 1667             2381  2
; 1668           P 2382  2 $FAB_INIT ( FAB = fab,
; 1669           P 2383  2             FNS = .filename [dsc$w_length],
; 1670           P 2384  2             FNA = .filename [dsc$a_pointer],
; 1671           P 2385  2             DNS = .syshelp [0],
; 1672           P 2386  2             DNA = syshelp [1],
; 1673             2387  2             NAM = nam);
; 1674             2388  2
; 1675             2389  3 IF (status = $PARSE (FAB = fab))
; 1676             2390  2    THEN (status = $SEARCH (FAB = fab));
; 1677             2391  2
; 1678             2392  2 RETURN .status;
; 1679             2393  1 END;


                                                                   .EXTRN  SYS$SEARCH

                                                 003C 00000 FILE_PRESENT:
                                                                   .WORD   Save R2,R3,R4,R5                              ; 2346
                          5E      FE50   CE  9E 00002              MOVAB   -432(SP), SP                                  ;
```

```
    0060  8F        .  00          6E          00  2C 00007         MOVC5   #0, (SP), #0, #96, $RMS_PTR                    ; 2380
                                         FF50  CD     0000E                                                                   :
                     FF50  CD          6002  8F  B0 00011         MOVW    #24578, $RMS_PTR                                     :
                     FF5A  CD                01  8E 00018         MNEGB   #1, $RMS_PTR+10                                      :
                     FF5C  CD                6E  9E 0001D         MOVAB   STRING, $RMS_PTR+12                                  :
    0050  8F           00          6E          00  2C 00022         MOVC5   #0, (SP), #0, #80, $RMS_PTR                    ; 2387
                                           B0  AD     00029                                                                   :
                       B0  AD          5003  8F  B0 0002B         MOVW    #20483, $RMS_PTR                                     :
                       C6  AD                02  90 00031         MOVB    #2, $RMS_PTR+22                                      :
                       CF  AD                02  90 00035         MOVB    #2, $RMS_PTR+31                                      :
                       D8  AD          FF50  CD  9E 00039         MOVAB   NAM, $RMS_PTR+40                                     :
                           50            04  AC  D0 0003F         MOVL    FILENAME, R0                                         :
                       DC  AD            04  A0  D0 00043         MOVL    4(R0), $RMS_PTR+44                                   :
                       E0  AD          0000' CF  9E 00048         MOVAB   SYSHELP+1, $RMS_PTR+48                               :
                       E4  AD                60  90 0004E         MOVB    (R0), $RMS_PTR+52                                    :
                       E5  AD          0000' CF  90 00052         MOVB    SYSHELP, $RMS_PTR+53                                 :
                                           B0  AD  9F 00058         PUSHAB  FAB                                            ; 2389
                00000000G  00                01  FB 0005B         CALLS   #1, SYS$PARSE                                        :
                           0A                50  E9 00062         BLBC    STATUS, 1$                                           :
                                           B0  AD  9F 00065         PUSHAB  FAB                                            ; 2390
                00000000G  00                01  FB 00068         CALLS   #1, SYS$SEARCH                                       :
                                                 04 0006F 1$:     RET                                                      ; 2393

; Routine Size:  112 bytes,    Routine Base:  $CODE$ + 0CEF
```

```
  1 %SBTTL 'Routine nohelp_log';
  1 ROUTINE nohelp_log (logdesc) =
  2 BEGIN
  2
  2 !++
  2 ! FUNCTIONAL DESCRIPTION:
  2 !
  2 !       If the logical name for a log file is defined, then put a record
  2 !       into the specified log file.  If that file does not already exist,
  2 !       then create it.
  2 !
  2 ! INPUTS:
  2 !
  2 !       logdesc = address of string descriptor for record to be output
  2 !
  2 ! OUTPUTS:
  2 !
  2 !       None.
  2 !
  2 ! ROUTINE VALUE:
  2 !
  2 !       Always true.
  2 !
  2 !--
  2
  2 MAP
  2     logdesc : REF BBLOCK;                           !Descriptor of log record
  2
  2 LOCAL
  2     logresult  : VECTOR [nam$c_maxrss, BYTE],       !Space for HELP$LOG resultant filename
  2     logrsdesc  : BBLOCK [dsc$c_s_bln],              !Descriptor for result name
  2     lognam     : BBLOCK [nam$c_bln],                !NAM block for HELP$LOG
  2     logfab     : BBLOCK [fab$c_bln],                !FAB for output to HELP$LOG
  2     lograb     : BBLOCK [rab$c_bln],                !RAB for output to HELP$LOG
  2     logfile    : BBLOCK [dsc$c_s_bln],
  2     logfiletrn : BBLOCK [dsc$c_s_bln],              !Descriptor for HELP$LOG translation
  2     status;
  2
  2 OWN
  2     logstring : COUNTEDSTRING ('HELP$LOG');
  2
  2 logfile [dsc$w_length] = .logstring [0];           ! Initialize logical name
  2 logfile [dsc$a_pointer] = logstring [1];
  2
  2 logfiletrn [dsc$w_length] = nam$c_maxrss;          ! Initialize descriptor for logical
  2 logfiletrn [dsc$a_pointer] = logresult;            ! name translation
  2
P 2 $NAM_INIT ( NAM = lognam,                         ! Initialize name block
P 2             ESS = nam$c_maxrss,
P 2             ESA = logresult,
P 2             RSS = nam$c_maxrss,
  2             RSA = logresult);
  2
P 2 $FAB_INIT ( FAB = logfab,                         ! Initialize fab
P 2             FNS = .logfile [dsc$w_length],
P 2             FNA = .logfile [dsc$a_pointer],
P 2             FAC = PUT,
; 1738       P 2451  2              FOP = CIF,
; 1739       P 2452  2              RAT = CR,
; 1740         2453  2              NAM = lognam);
; 1741         2454  2
; 1742       P 2455  2 $RAB_INIT ( RAB = lograb,                              ! Initialize rab
; 1743       P 2456  2              FAB = logfab,
; 1744         2457  2              ROP = EOF);
; 1745         2458  2
; 1746         2459  2 !
; 1747         2460  2 ! If HELP$LOG can be successfully translated,
; 1748         2461  2 !        1.  Create the file if it doesn't already exist.
; 1749         2462  2 !        2.  Connect to that file.
; 1750         2463  2 !        3.  Write the record to that file.
; 1751         2464  2 !        4.  Clean up afterwards.
; 1752         2465  2 !
; 1753         2466  2
; 1754         2467  4 IF ((status = $TRNLOG (LOGNAM = logfile, RSLBUF = logfiletrn))
; 1755         2468  3    AND (.status NEQ SS$_NOTRAN))   THEN IF (status = $CREATE (FAB = logfab))
; 1756         2469  3         THEN BEGIN
; 1757         2470  4              IF (status = $CONNECT (RAB = lograb))
; 1758         2471  4                 THEN BEGIN
; 1759         2472  4                      lograb [rab$w_rsz] = .logdesc [dsc$w_length];
; 1760         2473  4                      lograb [rab$l_rbf] = .logdesc [dsc$a_pointer];
; 1761         2474  4                      logrsdesc [dsc$w_length] = .lognam [nam$b_rsl];
; 1762         2475  4                      logrsdesc [dsc$a_pointer] = .lognam [nam$l_rsa];
; 1763         2476  4                      $PUT (RAB = lograb);
; 1764         2477  5                      IF NOT (status = $DISCONNECT (RAB = lograb))
; 1765         2478  4                         THEN SIGNAL ((shr$_closeout OR hlp$c_facility OR sts$k_warning),
; 1766         2479  4                                      1, logrsdesc, .status, .lograb [rab$l_stv]);
; 1767         2480  3                      END;
; 1768         2481  4              IF NOT (status = $CLOSE (FAB= logfab))
; 1769         2482  3                 THEN SIGNAL ((shr$_closeout OR hlp$c_facility OR sts$k_warning),
; 1770         2483  3                              1, logrsdesc, .status, .lograb [rab$l_stv]);
; 1771         2484  2              END;
; 1772         2485  2 RETURN true
; 1773         2486  1 END;                                                   !Of nohelp_log


                                                              .PSECT  $OWN$,NOEXE,2

                                               000AB          .BLKB   1
                                          08   000AC LOGSTRING:
                                                              .BYTE   8                                         ;
                  47  4F  4C  24  50  4C  45  48  000AD       .ASCII  \HELP$LOG\                                ;

                                                              .EXTRN  SYS$CREATE, SYS$CONNECT
                                                              .EXTRN  SYS$PUT, SYS$DISCONNECT
                                                              .EXTRN  SYS$CLOSE

                                                              .PSECT  $CODE$,NOWRT,2

                                        007C  00000 NOHELP_LOG:
                                                              .WORD   Save R2,R3,R4,R5,R6                       ; 2395
                         56 00000000G  00  9E  00002          MOVAB   LIB$SIGNAL, R6                            ;
                         5E      FDF4  CE  9E  00009          MOVAB   -524(SP), SP                              ;
                    08   AE     0000'  CF  9B  0000E          MOVZBW  LOGSTRING, LOGFILE                        ; 2435
```

```
PAC
Sym
LBR

PSE
---
.
$GL
$CC

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

The
564
The
63
0 p

Mac
---
_$2

0 C

The

MAC
```

```
                         0C   AE      0000'  CF  9E 00014          MOVAB   LOGSTRING+1, LOGFILE+4                  ; 2436
                              6E         FF  8F  9B 0001A          MOVZBW  #255, LOGFILETRN                        ; 2438
                         04   AE       FF00  CD  9E 0001E          MOVAB   LOGRESULT, LOGFILETRN+4                 ; 2439
  0060    8F        00        6E             00  2C 00024          MOVC5   #0, (SP), #0, #96, $RMS_PTR             ; 2445
                                       00A4  CE     0002B                                                          ;
                       C0A4   CE       6002  8F  B0 0002E          MOVW    #24578, $RMS_PTR                        ;
                       00A6   CE             01  8E 00035          MNEGB   #1, $RMS_PTR+2                          ;
                       00A8   CE       FF00  CD  9E 0003A          MOVAB   LOGRESULT, $RMS_PTR+4                   ;
                       00AE   CE             01  8E 00041          MNEGB   #1, $RMS_PTR+10                         ;
                       00B0   CE       FF00  CD  9E 00046          MOVAB   LOGRESULT, $RMS_PTR+12                  ;
  0050    8F        00        6E             00  2C 0004D          MOVC5   #0, (SP), #0, #80, $RMS_PTR             ; 2453
                                         54  AE     00054                                                          ;
                         54   AE       5003  8F  B0 00056          MOVW    #20483, $RMS_PTR                        ;
                         58   AE   02000000  8F  D0 0005C          MOVL    #33554432, $RMS_PTR+4                   ;
                         6A   AE             01  90 00064          MOVB    #1, $RMS_PTR+22                         ;
                         72   AE       0202  8F  B0 00068          MOVW    #514, $RMS_PTR+30                       ;
                         7C   AE       00A4  CE  9E 0006E          MOVAB   LOGNAM, $RMS_PTR+40                     ;
                       0080   CE         0C  AE  D0 00074          MOVL    LOGFILE+4, $RMS_PTR+44                  ;
                       0088   CE         08  AE  90 0007A          MOVB    LOGFILE, $RMS_PTR+52                    ;
  0044    8F        00        6E             00  2C 00080          MOVC5   #0, (SP), #0, #68, $RMS_PTR             ; 2457
                                         10  AE     00087                                                          ;
                         10   AE       4401  8F  B0 00089          MOVW    #17409, $RMS_PTR                        ;
                         14   AE       0100  8F  3C 0008F          MOVZWL  #256, $RMS_PTR+4                        ;
                         4C   AE         54  AE  9E 00095          MOVAB   LOGFAB, $RMS_PTR+60                     ;
                                             7E  7C 0009A          CLRQ    -(SP)                                   ; 2467
                                             7E  D4 0009C          CLRL    -(SP)                                   ;
                                         0C  AE  9F 0009E          PUSHAB  LOGFILETRN                              ;
                                             7E  D4 000A1          CLRL    -(SP)                                   ;
                                         1C  AE  9F 000A3          PUSHAB  LOGFILE                                 ;
                  00000000G   00             06  FB 000A6          CALLS   #6, SYS$TRNLOG                          ;
                              52             50  D0 000AD          MOVL    R0, STATUS                              ;
                              09             52  E9 000B0          BLBC    STATUS, 1$                              ;
                  00000629    8F             52  D1 000B3          CMPL    STATUS, #1577                           ; 2468
                                             03  12 000BA          BNEQ    2$                                      ;
                                           008D  31 000BC  1$:     BRW     4$                                      ;
                                         54  AE  9F 000BF  2$:     PUSHAB  LOGFAB                                  ;
                  00000000G   00             01  FB 000C2          CALLS   #1, SYS$CREATE                          ;
                              52             50  D0 000C9          MOVL    R0, STATUS                              ;
                              7D             52  E9 000CC          BLBC    STATUS, 4$                              ;
                                         10  AE  9F 000CF          PUSHAB  LOGRAB                                  ; 2470
                  00000000G   00             01  FB 000D2          CALLS   #1, SYS$CONNECT                         ;
                              52             50  D0 000D9          MOVL    R0, STATUS                              ;
                              49             52  E9 000DC          BLBC    STATUS, 3$                              ;
                              50         04  AC  D0 000DF          MOVL    LOGDESC, R0                             ; 2472
                         32   AE             60  B0 000E3          MOVW    (R0), LOGRAB+34                         ;
                         38   AE         04  A0  D0 000E7          MOVL    4(R0), LOGRAB+40                        ; 2473
                       0104   CE       00A7  CE  9B 000EC          MOVZBW  LOGNAM+3, LOGRSDESC                     ; 2474
                       FEFC   CD       00A8  CE  D0 000F3          MOVL    LOGNAM+4, LOGRSDESC+4                   ; 2475
                                         10  AE  9F 000FA          PUSHAB  LOGRAB                                  ; 2476
                  00000000G   00             01  FB 000FD          CALLS   #1, SYS$PUT                             ;
                                         10  AE  9F 00104          PUSHAB  LOGRAB                                  ; 2477
                  00000000G   00             01  FB 00107          CALLS   #1, SYS$DISCONNECT                      ;
                              52             50  D0 0010E          MOVL    R0, STATUS                              ;
                              14             52  E8 00111          BLBS    STATUS, 3$                              ;
                                         1C  AE  DD 00114          PUSHL   LOGRAB+12                               ; 2479
                                             52  DD 00117          PUSHL   STATUS                                  ;
                                       FEF8  CD  9F 00119          PUSHAB  LOGRSDESC                               ; 2478
                                       01  DD 0001D          PUSHL   #1                            ;
                          00761058     8F  DD 0011F          PUSHL   #7737432                      ;
                    66                 05  FB 00125          CALLS   #5, LIB$SIGNAL                ;
                                 54    AE  9F 00128 3$:      PUSHAB  LOGFAB                        ; 2481
          00000000G 00                 01  FB 0012B          CALLS   #1, SYS$CLOSE                 ;
                    52                 50  D0 00132          MOVL    R0, STATUS                    ;
                    14                 52  E8 00135          BLBS    STATUS, 4$                    ;
                                 1C    AE  DD 00138          PUSHL   LOGRAB+12                     ; 2483
                                       52  DD 0013B          PUSHL   STATUS                        ;
                               FEF8    CD  9F 0013D          PUSHAB  LOGRSDESC                     ; 2482
                                       01  DD 00141          PUSHL   #1                            ;
                          00761058     8F  DD 00143          PUSHL   #7737432                      ;
                    66                 05  FB 00149          CALLS   #5, LIB$SIGNAL                ;
                    50                 01  D0 0014C 4$:      MOVL    #1, R0                        ; 2485
                                           04 0014F          RET                                   ; 2486

; Routine Size:  336 bytes,    Routine Base:  $CODE$ + 0D5F
; 1775         2487  1 %SBTTL 'Routine remove_last_key';
; 1776         2488  1 ROUTINE remove_last_key (stringdescr, last_key_length) =
; 1777         2489  2 BEGIN
; 1778         2490  2
; 1779         2491  2 !++
; 1780         2492  2 ! FUNCTIONAL DESCRIPTION:
; 1781         2493  2 !
; 1782         2494  2 !       Remove the last keyword in the supplied string descriptor.
; 1783         2495  2 !
; 1784         2496  2 ! INPUTS:
; 1785         2497  2 !
; 1786         2498  2 !       stringdescr =   address of string descriptor for input text string
; 1787         2499  2 !
; 1788         2500  2 !       last_key_length = the length of the key that is to be removed
; 1789         2501  2 !
; 1790         2502  2 ! OUTPUTS:
; 1791         2503  2 !
; 1792         2504  2 !       stringdescr =   input descriptor with last key removed
; 1793         2505  2 !
; 1794         2506  2 ! ROUTINE VALUE:
; 1795         2507  2 !
; 1796         2508  2 !       Always true.
; 1797         2509  2 !
; 1798         2510  2 !--
; 1799         2511  2
; 1800         2512  2 MAP
; 1801         2513  2     stringdescr : REF BBLOCK;
; 1802         2514  2
; 1803         2515  2 LOCAL
; 1804         2516  2     last_char;                                          ! Pointer to last character of the last key
; 1805         2517  2
; 1806         2518  2 last_char = .stringdescr [dsc$w_length]                ! Find last char before " subtopic? "
; 1807         2519  2           + .stringdescr [dsc$a_pointer] - 12
; 1808         2520  2           - .last_key_length;
; 1809         2521  2
; 1810         2522  2 CH$MOVE (.subtopic [0], subtopic [1], .last_char + 1);  ! Replace "subtopic? " string
; 1811         2523  2 stringdescr [dsc$w_length] = .last_char + 1)          ! Calculate new length
; 1812         2524  2                         - .stringdescr [dsc$a_pointer];
; 1813         2525  2
; 1814         2526  2 RETURN true;
; 1815         2527  1 END;                                                   ! Of remove_last_key



                                               00FC 00000 REMOVE_LAST_KEY:
                                                               .WORD   Save R2,R3,R4,R5,R6,R7                       ; 2488
                       57       04   AC   D0 00002             MOVL    STRINGDESCR, R7                              ; 2518
                       56            67   3C 00006             MOVZWL  (R7), R6                                     ; 2519
                       56       04   A7   C0 00009             ADDL2   4(R7), R6                                    ;
                       56       08   AC   C2 0000D             SUBL2   LAST_KEY_LENGTH, R6                          ; 2520
                       56            0C   C2 00011             SUBL2   #12, LAST_CHAR                               ;
                       50     0000'  CF   9A 00014             MOVZBL  SUBTOPIC, R0                                 ; 2522
      01    A6    0000'  CF          50   28 00019             MOVC3   R0, SUBTOPIC+1, 1(LAST_CHAR)                 ;
                       56       04   A7   C2 00020             SUBL2   4(R7), R6                                    ; 2524
            67         56            0B   A1 00024             ADDW3   #11, R6, (R7)                                ;
```

```
                        50          01  D0 00028          MOVL    #1, R0                                                  ; 2526
                                        04 0002B          RET                                                             ; 2527

; Routine Size:  44 bytes,    Routine Base:  $CODE$ + 0EAF
; 1817          2528  1 %SBTTL 'Routine remove_terminator';
; 1818          2529  1 ROUTINE remove_terminator (stringdescr) =
; 1819          2530  2 BEGIN
; 1820          2531  2
; 1821          2532  2 !++
; 1822          2533  2 ! FUNCTIONAL DESCRIPTION:
; 1823          2534  2 !
; 1824          2535  2 !       Remove the termination characters at the end of the string descriptor.
; 1825          2536  2 !
; 1826          2537  2 ! INPUTS:
; 1827          2538  2 !
; 1828          2539  2 !       stringdescr =   address of string descriptor for input text string
; 1829          2540  2 !
; 1830          2541  2 ! OUTPUTS:
; 1831          2542  2 !
; 1832          2543  2 !       stringdescr =   input descriptor with termination characters removed
; 1833          2544  2 !
; 1834          2545  2 ! ROUTINE VALUE:
; 1835          2546  2 !
; 1836          2547  2 !       Always true.
; 1837          2548  2 !
; 1838          2549  2 !--
; 1839          2550  2
; 1840          2551  2 MAP
; 1841          2552  2     stringdescr : REF BBLOCK;
; 1842          2553  2
; 1843          2554  3 WHILE (CH$RCHAR (.stringdescr [dsc$a_pointer]                ! While termination character present
; 1844          2555  3               + .stringdescr [dsc$w_length] - 1) LSS %X'20')
; 1845          2556  3       AND (.stringdescr [dsc$w_length] GTR 0)
; 1846          2557  2 DO stringdescr [dsc$w_length] = .stringdescr [dsc$w_length] - 1;! Remove terminator
; 1847          2558  2
; 1848          2559  2 RETURN true;
; 1849          2560  1 END;                                                         ! Of remove_terminator


                                          0000 00000 REMOVE_TERMINATOR:
                                                              .WORD   Save nothing                          : 2529
                     51      04   AC   D0 00002               MOVL    STRINGDESCR, R1                       : 2554
                     50      04   BC   3C 00006 1$:           MOVZWL  @STRINGDESCR, R0                      : 2555
                     50      04   A1   C0 0000A               ADDL2   4(R1), R0                             :
                     20      FF   A0   91 0000E               CMPB    -1(R0), #32                           :
                                  0A   1E 00012               BGEQU   2$                                    :
                             04   BC   B5 00014               TSTW    @STRINGDESCR                          : 2556
                                  05   13 00017               BEQL    2$                                    :
                             04   BC   B7 00019               DECW    @STRINGDESCR                          : 2557
                                  E8   11 0001C               BRB     1$                                    :
                     50           01   D0 0001E 2$:           MOVL    #1, R0                                : 2559
                                       04 00021               RET                                           : 2560

; Routine Size:  34 bytes,    Routine Base:  $CODE$ + 0EDB
```

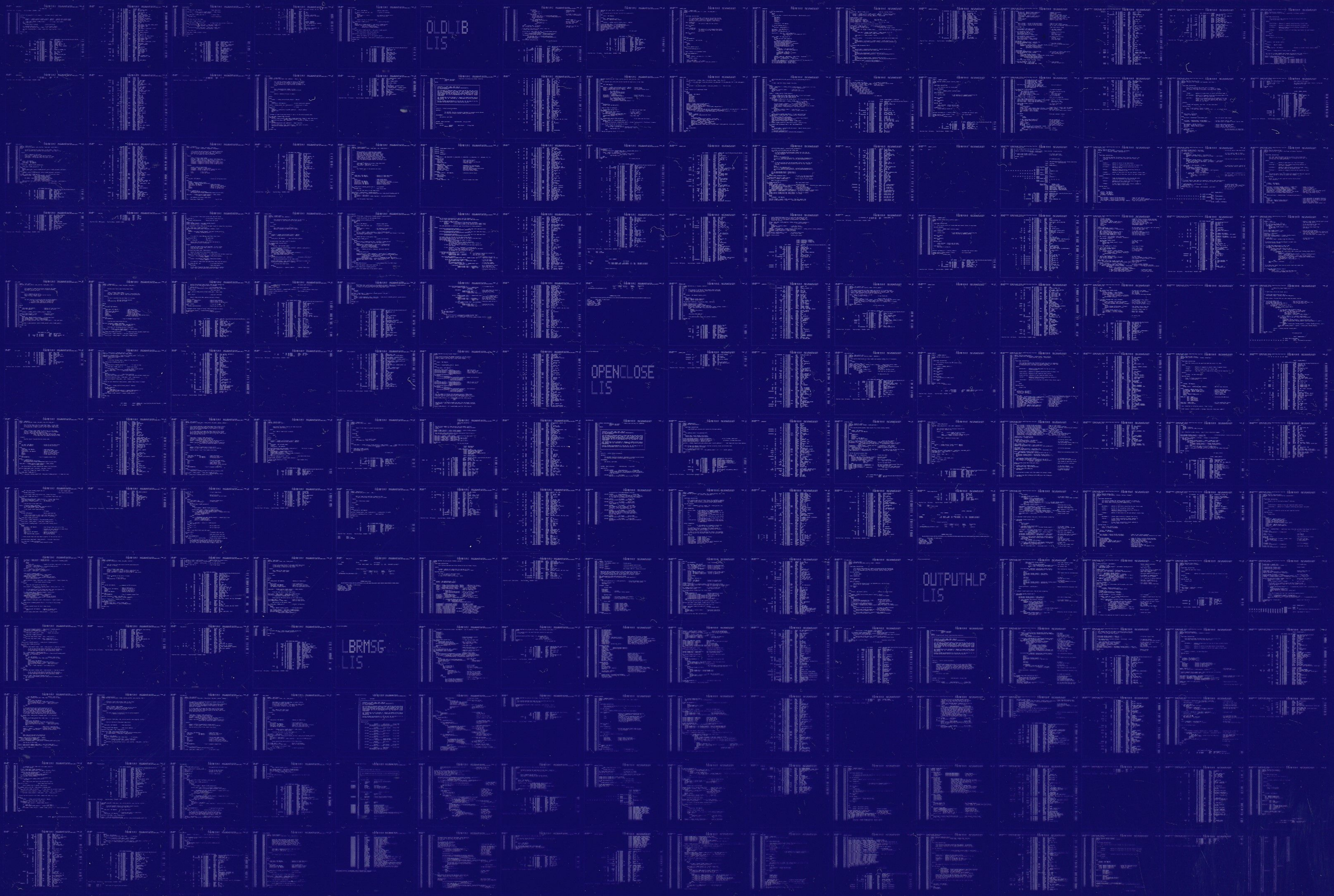
```
1 %SBTTL 'Routine make_upper_case';
1 ROUTINE make_upper_case (idesc: REF BLOCK[, BYTE], oname: REF VECTOR[, BYTE]) =
2 BEGIN
2
2 !++
2 ! FUNCTIONAL DESCRIPTION:
2 !
2 !       Upper case the name described by string descriptor idesc and
2 !       put the name at location oname.  (Also substitutes a space
2 !       character for a horizontal tab.)
2 !
2 ! INPUTS:
2 !
2 !       idesc =         address of string descriptor for input text string
2 !
2 !       oname =         address of buffer to contain uppercase output string
2 !
2 ! OUTPUTS:
2 !
2 !       oname : as described above
2 !
2 ! ROUTINE VALUE:
2 !
2 !       Always true.
2 !
2 !--
2
2 LITERAL
2     fill= 0;
2
2 OWN
2     upcase_table: VECTOR [256, BYTE] INITIAL (BYTE (
2 0, 1, 2, 3, 4, 5, 6, 7, 8, 32, 10, 11, 12, 13, 14, 15,          ! HT -> space.
2 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31,
2 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47,
2 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63,
2 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79,
2 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95,
2 96, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, ! a-o -> A-O.
2 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90,123,124,125,126,127, ! p-z -> P-Z.
2 128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,
2 144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,
2 160,161,162,163,164,165,166,167,168,169,170,171,172,173,174,175,
2 176,177,178,179,180,181,182,183,184,185,186,187,188,189,190,191,
2 192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,207,
2 208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,
2 192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,207, ! Upcase foreign
2 240,209,210,211,212,213,214,215,216,217,218,219,220,221,254,255)); ! too.
2
2 CH$TRANSLATE (upcase_table,
2     .idesc[dsc$w_length], .idesc[dsc$a_pointer], fill,
2     .idesc[dsc$w_length], .oname);
2
2 RETURN true
1 END;                                   !Of make_uppercase
```

```
                                                                              .PSECT  $OWN$,NOEXE,2

                                                                     000B5    .BLKB   3
 0E  0D  0C  0B  0A  20  08  07  06  05  04  03  02  01  00  000B8 UPCASE_TABLE:
                                                                              .BYTE   0, 1, 2, 3, 4, 5, 6, 7, 8, 32, 10, 11, -            ;
 1D  1C  1B  1A  19  18  17  16  15  14  13  12  11  10  0F  000C7                    12, 13, 14, 15, 16, 17, 18, 19, 20, 21, -          ;
 2C  2B  2A  29  28  27  26  25  24  23  22  21  20  1F  1E  000D6                    22, 23, 24, 25, 26, 27, 28, 29, 30, 31, -          ;
 3B  3A  39  38  37  36  35  34  33  32  31  30  2F  2E  2D  000E5                    32, 33, 34, 35, 36, 37, 38, 39, 40, 41, -          ;
 4A  49  48  47  46  45  44  43  42  41  40  3F  3E  3D  3C  000F4                    42, 43, 44, 45, 46, 47, 48, 49, 50, 51, -          ;
 59  58  57  56  55  54  53  52  51  50  4F  4E  4D  4C  4B  00103                    52, 53, 54, 55, 56, 57, 58, 59, 60, 61, -          ;
 48  47  46  45  44  43  42  41  60  5F  5E  5D  5C  5B  5A  00112                    62, 63, 64, 65, 66, 67, 68, 69, 70, 71, -          ;
 57  56  55  54  53  52  51  50  4F  4E  4D  4C  4B  4A  49  00121                    72, 73, 74, 75, 76, 77, 78, 79, 80, 81, -          ;
 86  85  84  83  82  81  80  7F  7E  7D  7C  7B  5A  59  58  00130                    82, 83, 84, 85, 86, 87, 88, 89, 90, 91, -          ;
 95  94  93  92  91  90  8F  8E  8D  8C  8B  8A  89  88  87  0013F                    92, 93, 94, 95, 96, 65, 66, 67, 68, 69, -          ;
 A4  A3  A2  A1  A0  9F  9E  9D  9C  9B  9A  99  98  97  96  0014E                    70, 71, 72, 73, 74, 75, 76, 77, 78, 79, -          ;
 B3  B2  B1  B0  AF  AE  AD  AC  AB  AA  A9  A8  A7  A6  A5  0015D                    80, 81, 82, 83, 84, 85, 86, 87, 88, 89, -          ;
 C2  C1  C0  BF  BE  BD  BC  BB  BA  B9  B8  B7  B6  B5  B4  0016C                    90, 123, 124, 125, 126, 127, -128, -127, -         ;
 D1  D0  CF  CE  CD  CC  CB  CA  C9  C8  C7  C6  C5  C4  C3  0017B                    -126, -125, -124, -123, -122, -121, -120, -        ;
 C0  DF  DE  DD  DC  DB  DA  D9  D8  D7  D6  D5   4  D3  D2  0018A                    -119, -118, -117, -116, -115, -114, -113, -        ;
 CF  CE  CD  CC  CB  CA  C9  C8  C7  C6  C5  C4  C3  C2  C1  00199                    -112, -111, -110, -109, -108, -107, -106, -        ;
 FE  DD  DC  DB  DA  D9  D8  D7  D6  D5  D4  D3  D2  D1  F0  001A8                    -105, -104, -103, -102, -101, -100, -99, -         ;
                                                         FF  001B7                    -98, -97, -96, -95, -94, -93, -92, -91, -          ;
                                                                                      -90, -89, -88, -87, -86, -85, -84, -83, -          ;
                                                                                      -82, -81, -80, -79, -78, -77, -76, -75, -          ;
                                                                                      -74, -73, -72, -71, -70, -69, -68, -67, -          ;
                                                                                      -66, -65, -64, -63, -62, -61, -60, -59, -          ;
                                                                                      -58, -57, -56, -55, -54, -53, -52, -51, -          ;
                                                                                      -50, -49, -48, -47, -46, -45, -44, -43, -          ;
                                                                                      -42, -41, -40, -39, -38, -37, -36, -35, -          ;
                                                                                      -34, -33, -64, -63, -62, -61, -60, -59, -          ;
                                                                                      -58, -57, -56, -55, -54, -53, -52, -51, -          ;
                                                                                      -50, -49, -16, -47, -46, -45, -44, -43, -          ;
                                                                                      -42, -41, -40, -39, -38, -37, -36, -35, -          ;
                                                                                      -2, -1                                             ;


                                                                              .PSECT  $CODE$,NOWRT,2

                                                           003C 00000 MAKE_UPPER_CASE:
                                                                              .WORD   Save R2,R3,R4,R5                                   ; 2562
                                   50        04    AC  D0 00002               MOVL    IDESC, R0                                          ; 2611
     0000'  CF          00     04  B0              60  2E 00006               MOVTC   (R0), @4(R0), #0, UPCASE_TABLE, (R0), -            ; 2612
                               08  BC              60     0000E                       @DNAME                                             ;
                                   50              01  D0 00011               MOVL    #1, R0                                             ; 2614
                                                       04 00014               RET                                                        ; 2615

; Routine Size:  21 bytes,    Routine Base:  $CODE$ + 0EFD


; 1906          2616   0 END ELUDOM



                                                                              .EXTRN  LIB$SIGNAL
```

```
;                                       PSECT SUMMARY
;
;       Name                        Bytes                         Attributes
;
;   $OWN$                              440  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                            3858  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)




;                                  Library Statistics
;
;                                                 -------- Symbols --------      Pages       Processing
;       File                                      Total    Loaded   Percent     Mapped      Time
;
;   _$255$DUA28:[SYSLIB]STARLET.L32;1               9776       127         1       581         00:01.0




;                                        COMMAND QUALIFIERS

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

; Size:          3858 code + 440 data bytes
; Run Time:         01:15.5
; Elapsed Time:     02:32.9
; Lines/CPU Min:     2079
; Lexemes/CPU-Min: 27990
; Memory Used:  351 pages
; Compilation Complete
```

0199 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



0200 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

TRANSFER
LIS

DATABASE
LIS

PUTCACHE
LIS

LIBRAR

PREFIX
REQ

LIBRARIAN
MAP

CROSS
LIS

SUBS
LIS

FILEIO
LIS

PADLBR
LIS

COMPRESS
LIS

EXTRACT
LIS

LIB
MDL

DELETE
LIS