```
HHH          HHH  EEEEEEEEEEEEEEE  LLL              PPPPPPPPPPPP
HHH          HHH  EEEEEEEEEEEEEEE  LLL              PPPPPPPPPPPPP
HHH          HHH  EEEEEEEEEEEEEEE  LLL              PPPPPPPPPPPPP
HHH          HHH  EEE              LLL              PPP          PPP
HHH          HHH  EEE              LLL              PPP          PPP
HHH          HHH  EEE              LLL              PPP          PPP
HHH          HHH  EEE              LLL              PPP          PPP
HHH          HHH  EEE              LLL              PPP          PPP
HHH          HHH  EEE              LLL              PPP          PPP
HHHHHHHHHHHHHHHH  EEEEEEEEEEEE     LLL              PPPPPPPPPPPPP
HHHHHHHHHHHHHHHH  EEEEEEEEEEEE     LLL              PPPPPPPPPPPPP
HHHHHHHHHHHHHHHH  EEEEEEEEEEEE     LLL              PPPPPPPPPPPPP
HHH          HHH  EEE              LLL              PPP
HHH          HHH  EEE              LLL              PPP
HHH          HHH  EEE              LLL              PPP
HHH          HHH  EEE              LLL              PPP
HHH          HHH  EEE              LLL              PPP
HHH          HHH  EEE              LLL              PPP
HHH          HHH  EEEEEEEEEEEEEEE  LLLLLLLLLLLLLLL  PPP
HHH          HHH  EEEEEEEEEEEEEEE  LLLLLLLLLLLLLLL  PPP
HHH          HHH  EEEEEEEEEEEEEEE  LLLLLLLLLLLLLLL  PPP
```

FILEID**HELP

```
HH      HH  EEEEEEEEEE  LL          PPPPPPPP
HH      HH  EEEEEEEEEE  LL          PPPPPPPP
HH      HH  EE          LL          PP      PP
HH      HH  EE          LL          PP      PP
HH      HH  EE          LL          PP      PP
HH      HH  EE          LL          PP      PP
HHHHHHHHHH  EEEEEEEE    LL          PPPPPPPP
HHHHHHHHHH  EEEEEEEE    LL          PPPPPPPP
HH      HH  EE          LL          PP
HH      HH  EE          LL          PP
HH      HH  EE          LL          PP              ....
HH      HH  EE          LL          PP              ....
HH      HH  EEEEEEEEEE  LLLLLLLLLL  PP              ....
HH      HH  EEEEEEEEEE  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
```

```
;     1          0001  0 MODULE help_help (
;     2          0002  0                 LANGUAGE (BLISS32),
;     3          0003  0                 ADDRESSING_MODE(EXTERNAL=GENERAL,
;     4          0004  0                                 NONEXTERNAL=LONG_RELATIVE),
;     5          0005  0                 IDENT = 'V04-000',
;     6          0006  0                 MAIN = HELP$START
;     7          0007  0                 ) =
;     8          0008  1 BEGIN
;     9          0009  1
;    10          0010  1 !
;    11          0011  1 !*****************************************************************************
;    12          0012  1 !*                                                                           *
;    13          0013  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    14          0014  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    15          0015  1 !*  ALL RIGHTS RESERVED.                                                      *
;    16          0016  1 !*                                                                           *
;    17          0017  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    18          0018  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;    19          0019  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;    20          0020  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    21          0021  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    22          0022  1 !*  TRANSFERRED.                                                              *
;    23          0023  1 !*                                                                           *
;    24          0024  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    25          0025  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    26          0026  1 !*  CORPORATION.                                                              *
;    27          0027  1 !*                                                                           *
;    28          0028  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    29          0029  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    30          0030  1 !*                                                                           *
;    31          0031  1 !*                                                                           *
;    32          0032  1 !*****************************************************************************
;    33          0033  1
;    34          0034  1 !++
;    35          0035  1 !
;    36          0036  1 ! FACILITY:  DCL $HELP command
;    37          0037  1 !
;    38          0038  1 ! ABSTRACT:
;    39          0039  1 !
;    40          0040  1 !       The DCL HELP command provides on-line information retrieval.
;    41          0041  1 !
;    42          0042  1 ! ENVIRONMENT:
;    43          0043  1 !
;    44          0044  1 !       VAX native, user mode.
;    45          0045  1 !
;    46          0046  1 !--
;    47          0047  1 !
;    48          0048  1 !
;    49          0049  1 ! AUTHOR:  Peter George,         CREATION DATE:  1-May-1981
;    50          0050  1 !
;    51          0051  1 ! MODIFIED BY:
;    52          0052  1 !
;    53          0053  1 !       V03-006 MCN0177         Maria del C. Nasr       11-Jul-1984
;    54          0054  1 !               Make /NOOUTPUT suppress all output.  Allow prompting
;    55          0055  1 !               with /[NO]OUTPUT qualifier only if explicitly specified.
;    56          0056  1 !
;    57          0057  1 !       V03-005 PCG0017         Peter George            11-Apr-1984
```

```
;    58          0058  1 !
;    59          0059  1 !               Modify defaulting of output stream, processing
;    60          0060  1 !               of /USERLIBRARY, and signalling of errors returned
;    61          0061  1 !               by LBR$OUTPUT_HELP.
;    62          0062  1 !
;    63          0063  1 !       V03-004 PCG0016         Peter George            12-Oct-1983
;    64          0064  1 !               Abort on write errors.
;    65          0065  1 !
;    66          0066  1 !       V03-003 PCG0015         Peter George            15-Sep-1983
;    67          0067  1 !               Change page break prompt.
;    68          0068  1 !
;    69          0069  1 !       V03-002 PCG0014         Peter George            15-Dec-1982
;    70          0070  1 !               Use CLI$INTERFACE routines for parsing.
;    71          0071  1 !               Clean up some code.
;    72          0072  1 !               Fix HELP/PAGE in command procedures.
;    73          0073  1 !
;    74          0074  1 !       V03-001 PCG0013         Peter George            01-Jul-1982
;    75          0075  1 !--             Add /LIBLIST and /INSTRUCTION qualifiers.
```

```
;    77          0076  1 LIBRARY
;    78          0077  1     'SYS$LIBRARY:STARLET';
;    79          0078  1
;    80          0079  1 REQUIRE
;    81          0080  1     'HELPDEF';
;    82          0671  1
;    83          0672  1 !
;    84          0673  1 ! Declare static strings
;    85          0674  1 !
;    86          0675  1 MACRO
;    87        M 0676  1     SD[A] =
;    88          0677  1         BIND %NAME('SD_',A) = $descriptor(a)%;
;    89          0678  1
;    90        P 0679  1 SD(
;    91        P 0680  1         'KEYWORDS',
;    92        P 0681  1         'PROMPT',
;    93        P 0682  1         'PAGE',
;    94        P 0683  1         'OUTPUT',
;    95        P 0684  1         'LIBRARY',
;    96        P 0685  1         'LIBLIST',
;    97        P 0686  1         'INSTRUCTIONS',
;    98        P 0687  1         'USERLIBRARY',
;    99        P 0688  1         'ALL',
;   100        P 0689  1         'NONE',
;   101        P 0690  1         'PROCESS',
;   102        P 0691  1         'GROUP',
;   103        P 0692  1         'SYSTEM',
;   104          0693  1         );
;   105          0694  1
;   106          0695  1 BIND
;   107          0696  1     pagebrk_prompt = $DESCRIPTOR ('Press RETURN to continue ... '),
;   108          0697  1     sysinput = $DESCRIPTOR ('SYS$INPUT') : $BBLOCK;
;   109          0698  1
;   110          0699  1 EXTERNAL ROUTINE
;   111          0700  1     lib$get_input,              ! Get a line from SYS$INPUT
;   112          0701  1     scr$erase_page,             ! Clear screen
;   113          0702  1     lbr$output_help,            ! Get help text
;   114          0703  1     cli$get_value,              ! Get entity value
;   115          0704  1     cli$present;                ! Is entity present
;   116          0705  1
;   117          0706  1 EXTERNAL
;   118          0707  1     lbr$gl_rmsstv : ADDRESSING_MODE(GENERAL);   ! RMS STV from librarian
;   119          0708  1
;   120          0709  1 FORWARD ROUTINE
;   121          0710  1     get_input,                                  ! Get a line of input text
;   122          0711  1     print_help_line,                            ! Driver to call help output routines
;   123          0712  1     put_page_break,                             ! Put a page break
;   124          0713  1     put_output,                                 ! Put a line of output text to the screen
;   125          0714  1     find_file_info,                             ! Determine characteristics of SYS$OUPUT
;   126          0715  1     open_sysinput,                              ! Open SYS$INPUT
;   127          0716  1     open_sysoutput,                             ! Open SYS$OUTPUT
;   128          0717  1     clean_up,                                   ! Deassign, disconnect, close all open files
;   129          0718  1     make_upper_case;                            ! Upper case a string
;   130          0719  1
;   131          0720  1 EXTERNAL LITERAL
;   132          0721  1     cli$_present,                               ! Entity is present
;   133          0722  1     cli$_defaulted,                             ! Entity is present by default
```

```
;  134          0723  1      cli$_negated,                    ! Entity is explicitly negated
;  135          0724  1      lbr$_endtopic;                   ! Status telling lbr$output_help to abort
;  136          0725  1                                       ! help on a particular topic
;  137          0726  1
;  138          0727  1  LITERAL
;  139          0728  1      true = 1,                        ! Truthness
;  140          0729  1      false = 0;                       ! Falsity
;  141          0730  1
;  142          0731  1  OWN
;  143          0732  1      sysinchan,                       ! Channel assigned to SYS$INPUT
;  144          0733  1      sysoutrab  : $BBLOCK [rab$c_bln],  ! RAB for output to SYS$OUTPUT
;  145          0734  1      sysout_name : $BBLOCK [dsc$c_s_bln],  ! String descriptor for result name
;  146          0735  1      sysoutdesc : $BBLOCK [dsc$c_s_bln],  ! String descriptor for output file name
;  147          0736  1      outputdesc : $BBLOCK [dsc$c_s_bln],  ! Local descriptor for prompt response
;  148          0737  1      outputbuf : $BBLOCK [hlp$c_pagesize],  ! Buffer for prompt
;  149          0738  1      current_height : INITIAL (0),    ! Number of lines currently output
;  150          0739  1      list_height,                     ! Number of lines on a page
;  151          0740  1      help_flags : $BBLOCK [4];        ! Control flags
;  152          0741  1
```

```
;  154     0742  1 GLOBAL ROUTINE help$start (arglist) =
;  155     0743  2 BEGIN
;  156     0744  2
;  157     0745  2 !++
;  158     0746  2 ! FUNCTIONAL DESCRIPTION:
;  159     0747  2 !
;  160     0748  2 !       This routine is called by CLI when the HELP command is entered.
;  161     0749  2 !       The keys are parsed and the librarian is called to extract the
;  162     0750  2 !       help from the help library.
;  163     0751  2 !
;  164     0752  2 ! INPUTS:
;  165     0753  2 !
;  166     0754  2 !       User's command line.
;  167     0755  2 !
;  168     0756  2 ! OUTPUTS:
;  169     0757  2 !
;  170     0758  2 !       The requested help text is displayed on the current SYS$OUTPUT:
;  171     0759  2 !
;  172     0760  2 ! ROUTINE VALUE:
;  173     0761  2 !
;  174     0762  2 !       Always true.
;  175     0763  2 !
;  176     0764  2 !--
;  177     0765  2
;  178     0766  2 LOCAL
;  179     0767  2     getcmd_desc : $BBLOCK [dsc$c_s_bln],                    ! Command descriptor
;  180     0768  2     libraryname : $BBLOCK [dsc$c_s_bln],                    ! String descriptor for library name
;  181     0769  2     library_addr,                                           ! Address of library desc
;  182     0770  2     list_width,                                             ! Number of chars per line
;  183     0771  2     listingfab  : $BBLOCK [fab$c_bln],                      ! FAB for output to listing
;  184     0772  2     sysoutfab : $BBLOCK [fab$c_bln],                        ! FAB for output to terminal
;  185     0773  2     sysoutnam  : $BBLOCK [nam$c_bln],                       ! NAM block for SYS$OUTPUT
;  186     0774  2     status;
;  187     0775  2
;  188     0776  2 ! Get help keys and uppercase them
;  189     0777  2 !
;  190     0778  2 CH$FILL (0, dsc$c_s_bln, getcmd_desc);
;  191     0779  2 getcmd_desc [dsc$b_class] = dsc$k_class_d;
;  192     0780  2 CLI$GET_VALUE (sd_keywords, getcmd_desc);
;  193     0781  2 make_upper_case (getcmd_desc, .getcmd_desc [dsc$a_pointer]);
;  194     0782  2
;  195     0783  2 ! Get output file.
;  196     0784  2 !
;  197     0785  2 CH$FILL (0, dsc$c_s_bln, sysoutdesc);                       ! Init output descriptor
;  198     0786  2 IF CLI$PRESENT (sd_output)                                  ! If /OUTPUT
;  199     0787  2 THEN                                                        ! Then get output file
;  200     0788  3     BEGIN
;  201     0789  3     sysoutdesc [dsc$b_class] = dsc$k_class_d;
;  202     0790  3     CLI$GET_VALUE (sd_output, sysoutdesc)
;  203     0791  3     END
;  204     0792  2 ELSE
;  205     0793  2     ! /NOOUTPUT was specified, so suppress all output
;  206     0794  2     !
;  207     0795  3     BEGIN
;  208     0796  3     sysoutdesc [dsc$w_length] = %CHARCOUNT('NL:');
;  209     0797  3     sysoutdesc [dsc$a_pointer] = UPLIT BYTE ('NL:');
;  210     0798  2     END;
```

```
;   211          0799  2
;   212          0800  2 ! Get library file.
;   213          0801  2 !
;   214          0802  2 IF CLI$PRESENT (sd_library)                                  ! If /LIBRARY
;   215          0803  2 THEN                                                        ! Then get library name
;   216          0804  3     BEGIN
;   217          0805  3     CH$FILL (0, dsc$c_s_bln, libraryname);
;   218          0806  3     libraryname [dsc$b_class] = dsc$k_class_d;
;   219          0807  3     CLI$GET_VALUE (sd_library, libraryname);
;   220          0808  3     library_addr = libraryname;
;   221          0809  3     END
;   222          0810  2 ELSE library_addr = 0;                                       ! Else Use null arg in call
;   223          0811  2
;   224          0812  2 !
;   225          0813  2 ! Initialize all flags.  Set prompt flag off if /OUTPUT or /NOOUTPUT qualifier
;   226          0814  2 ! is present, and prompt set by default.
;   227          0815  2 !
;   228          0816  2 help_flags = 0;
;   229          0817  4 IF ((CLI$PRESENT (sd_output) EQL CLI$_PRESENT
;   230          0818  4      OR CLI$PRESENT (sd_output) EQL CLI$_NEGATED)
;   231          0819  3    AND (CLI$PRESENT (sd_prompt) EQL CLI$_DEFAULTED))
;   232          0820  2 THEN
;   233          0821  2     help_flags [hlp$v_prompt] = false
;   234          0822  2 ELSE
;   235          0823  2     help_flags [hlp$v_prompt] = CLI$PRESENT (sd_prompt);
;   236          0824  2
;   237          0825  2 !
;   238          0826  2 ! Parse /USERLIBRARY
;   239          0827  2 !
;   240          0828  2 help_flags = .help_flags AND NOT hlp$m_process                ! By default don't search any tables
;   241          0829  2                 AND NOT hlp$m_group AND NOT hlp$m_system;
;   242          0830  2 IF CLI$PRESENT (sd_userlibrary)                              ! If /USERLIBRARY is specified
;   243          0831  2 THEN
;   244          0832  3     BEGIN
;   245          0833  3     IF CLI$PRESENT (sd_all)                                 ! If ALL
;   246          0834  3     THEN help_flags = .help_flags OR hlp$m_process OR       ! Then search all tables
;   247          0835  3                       hlp$m_group OR hlp$m_system;
;   248          0836  3     IF CLI$PRESENT (sd_none)                                ! If NONE
;   249          0837  3     THEN help_flags = .help_flags AND NOT hlp$m_process     ! Then don't search any tables
;   250          0838  3                       AND NOT hlp$m_group AND NOT hlp$m_system;
;   251          0839  3     IF CLI$PRESENT (sd_process)                             ! If PROCESS
;   252          0840  3     THEN help_flags [hlp$v_process] = true;                 ! Then search it
;   253          0841  3     IF CLI$PRESENT (sd_group)                               ! If GROUP
;   254          0842  3     THEN help_flags [hlp$v_group] = true;                   ! Then search it
;   255          0843  3     IF CLI$PRESENT (sd_system)                              ! If SYSTEM
;   256          0844  3     THEN help_flags [hlp$v_system] = true;                  ! Then search it
;   257          0845  2     END;
;   258          0846  2
;   259          0847  2 !
;   260          0848  2 ! Get other flags
;   261          0849  2 !
;   262          0850  2 help_flags [hlp$v_page] = CLI$PRESENT (sd_page);            ! Set paging flag
;   263          0851  2 help_flags [hlp$v_help] = CLI$PRESENT (sd_instructions);    ! Set instructions flag
;   264          0852  2 help_flags [hlp$v_liblist] = CLI$PRESENT (sd_liblist);      ! Set default library list flag
;   265          0853  2
;   266          0854  2 !
;   267          0855  2 ! Get and set output device characteristics.
```

```
;  268          0856  2 !
;  269          0857  2 open_sysoutput (sysoutfab, sysoutnam);                    ! Open SYS$OUTPUT
;  270          0858  2 find_file_info (sysoutfab, list_width);                   ! Get characteristics of output
;  271          0859  2 current_height = .list_height - 3;                       ! Don't generate page break prompt yet
;  272          0860  2 outputdesc [dsc$w_length] = 0;                           ! Init prompt command descriptor
;  273          0861  2 outputdesc [dsc$a_pointer] = outputbuf;                  !
;  274          0862  2
;  275          0863  2 !
;  276          0864  2 ! Open SYS$INPUT if prompting or paging in effect
;  277          0865  2 !
;  278          0866  2 IF .help_flags [hlp$v_prompt] OR .help_flags [hlp$v_page]   ! If prompting or pating
;  279          0867  2    THEN open_sysinput ();                                  ! Then open SYS$INPUT
;  280          0868  2
;  281          0869  2 !
;  282          0870  2 ! Call lbr$output_help to do all the real work.
;  283          0871  2 !
;  284          0872  2 status = lbr$output_help (print_help_line, list_width,    ! Call LBR$OUTPUT_HELP
;  285          0873  2              getcmd_desc, .library_addr, help_flags, get_input); !
;  286          0874  2
;  287          0875  2 clean_up (sysoutfab);                                     ! Close all the files that have been opened
;  288          0876  2
;  289          0877  2 RETURN .status;
;  290          0878  1 END;                                                      ! Of help$start

                                                        .TITLE  HELP_HELP
                                                        .IDENT  \V04-000\

                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                    53  44  52  4F  57  59  45  4B  00000 P.AAB:  .ASCII  \KEYWORDS\                    ;
                                    00000008  00008 P.AAA:  .LONG   8                                  ;
                                    00000000' 0000C         .ADDRESS P.AAB                            ;
                            54  50  4D  4F  52  50  00010 P.AAD:  .ASCII  \PROMPT\                     ;
                                                00016         .BLKB   2                                 ;
                                    00000006  00018 P.AAC:  .LONG   6                                  ;
                                    00000000' 0001C         .ADDRESS P.AAD                            ;
                                        45  47  41  50  00020 P.AAF:  .ASCII  \PAGE\                   ;
                                    00000004  00024 P.AAE:  .LONG   4                                  ;
                                    00000000' 00028         .ADDRESS P.AAF                            ;
                            54  55  50  54  55  4F  0002C P.AAH:  .ASCII  \OUTPUT\                     ;
                                                00032         .BLKB   2                                 ;
                                    00000006  00034 P.AAG:  .LONG   6                                  ;
                                    00000000' 00038         .ADDRESS P.AAH                            ;
                        59  52  41  52  42  49  4C  0003C P.AAJ:  .ASCII  \LIBRARY\                    ;
                                                00043         .BLKB   1                                 ;
                                    00000007  00044 P.AAI:  .LONG   7                                  ;
                                    00000000' 00048         .ADDRESS P.AAJ                            ;
                        54  53  49  4C  42  49  4C  0004C P.AAL:  .ASCII  \LIBLIST\                    ;
                                                00053         .BLKB   1                                 ;
                                    00000007  00054 P.AAK:  .LONG   7                                  ;
                                    00000000' 00058         .ADDRESS P.AAL                            ;
53  4E  4F  49  54  43  55  52  54  53  4E  49  0005C P.AAN:  .ASCII  \INSTRUCTIONS\                   ;
                                    0000000C  00068 P.AAM:  .LONG   12                                 ;
                                    00000000' 0006C         .ADDRESS P.AAN                            ;
    59  52  41  52  42  49  4C  52  45  53  55  00070 P.AAP:  .ASCII  \USERLIBRARY\                    ;
                                                0007B         .BLKB   1
```

```
                                        00000000B  0007C P.AAO:   .LONG    11                                      ;
                                        00000000'  00080         .ADDRESS P.AAP                                   ;
                                     4C 4C 41      00084 P.AAR:   .ASCII   \ALL\                                   ;
                                                   00087         .BLKB    1                                       ;
                                        00000003   00088 P.AAQ:   .LONG    3                                       ;
                                        00000000'  0008C         .ADDRESS P.AAR                                   ;
                                  45 4E 4F 4E      00090 P.AAT:   .ASCII   \NONE\                                  ;
                                        00000004   00094 P.AAS:   .LONG    4                                       ;
                                        00000000'  00098         .ADDRESS P.AAT                                   ;
                           53 53 45 43 4F 52 50    0009C P.AAV:   .ASCII   \PROCESS\                               ;
                                                   000A3         .BLKB    1                                       ;
                                        00000007   000A4 P.AAU:   .LONG    7                                       ;
                                        00000000'  000A8         .ADDRESS P.AAV                                   ;
                                  50 55 4F 52 47   000AC P.AAX:   .ASCII   \GROUP\                                 ;
                                                   000B1         .BLKB    3                                       ;
                                        00000005   000B4 P.AAW:   .LONG    5                                       ;
                                        00000000'  000B8         .ADDRESS P.AAX                                   ;
                              4D 45 54 53 59 53    000BC P.AAZ:   .ASCII   \SYSTEM\                                ;
                                                   000C2         .BLKB    2                                       ;
                                        00000006   000C4 P.AAY:   .LONG    6                                       ;
                                        00000000'  000C8         .ADDRESS P.AAZ                                   ;
6F 74 20 4E 52 55 54 45 52 20 73 73 65 72 50       000CC P.ABB:   .ASCII   \Press RETURN to continue ... \          ;
   20 2E 2E 2E 20 65 75 6E 69 74 6E 6F 63 20       000DB                                                           ;
                                                   000E9         .BLKB    3                                       ;
                                        0000001D   000EC P.ABA:   .LONG    29                                      ;
                                        00000000'  000F0         .ADDRESS P.ABB                                   ;
                        54 55 50 4E 49 24 53 59 53 000F4 P.ABD:   .ASCII   \SYS$INPUT\                             ;
                                                   000FD         .BLKB    3                                       ;
                                        00000009   00100 P.ABC:   .LONG    9                                       ;
                                        00000000'  00104         .ADDRESS P.ABD                                   ;
                                     3A 4C 4E      00108 P.ABE:   .ASCII   \NL:\                                   ;

                                                                  .PSECT  $OWN$,NOEXE,2

                                                   00000 SYSINCHAN:
                                                                  .BLKB   4
                                                   00004 SYSOUTRAB:
                                                                  .BLKB   68
                                                   00048 SYSOUT_NAME:
                                                                  .BLKB   8
                                                   00050 SYSOUTDESC:
                                                                  .BLKB   8
                                                   00058 OUTPUTDESC:
                                                                  .BLKB   8
                                                   00060 OUTPUTBUF:
                                                                  .BLKB   512
                                        00000000   00260 CURRENT_HEIGHT:
                                                                  .LONG   0                                        ;
                                                   00264 LIST_HEIGHT:
                                                                  .BLKB   4
                                                   00268 HELP_FLAGS:
                                                                  .BLKB   4

                                                         SD_KEYWORDS=        P.AAA
                                                         SD_PROMPT=          P.AAC
                                                         SD_PAGE=            P.AAE
                                                         SD_OUTPUT=          P.AAG
```

```
                                        SD_LIBRARY=             P.AAI
                                        SD_LIBLIST=             P.AAK
                                        SD_INSTRUCTIONS=        P.AAM
                                        SD_USERLIBRARY=         P.AAO
                                        SD_ALL=                 P.AAQ
                                        SD_NONE=                P.AAS
                                        SD_PROCESS=             P.AAU
                                        SD_GROUP=               P.AAW
                                        SD_SYSTEM=              P.AAY
                                        PAGEBRK_PROMPT=         P.ABA
                                        SYSINPUT=               P.ABC
                                                .EXTRN  LIB$GET_INPUT, SCR$ERASE_PAGE
                                                .EXTRN  LBR$OUTPUT_HELP
                                                .EXTRN  CLI$GET_VALUE, CLI$PRESENT
                                                .EXTRN  LBR$GL_RMSSTV, CLI$_PRESENT
                                                .EXTRN  CLI$_DEFAULTED, CLI$_NEGATED
                                                .EXTRN  LBR$_ENDTOPIC

                                                .PSECT  $CODE$,NOWRT,2

                                  03FC 00000            .ENTRY  HELP$START, Save R2,R3,R4,R5,R6,R7,R8,R9    ; 0742
                59 00000000G   00   9E 00002            MOVAB   CLI$GET_VALUE, R9                           ;
                58 00000000G   00   9E 00009            MOVAB   CLI$PRESENT, R8                             ;
                57 00000000'   EF   9E 00010            MOVAB   SD_OUTPUT, R7                               ;
                56 00000000'   EF   9E 00017            MOVAB   HELP_FLAGS, R6                              ;
                5E     FEEC    CE   9E 0001E            MOVAB   -276(SP), SP                                ;
     08         00             6E   00   2C 00023      MOVC5   #0, (SP), #0, #8, GETCMD_DESC               ; 0778
                       F8      AD   00028              ;
                   FB  AD          02   90 0002A       MOVB    #2, GETCMD_DESC+3                           ; 0779
                       F8  AD      9F   0002E          PUSHAB  GETCMD_DESC                                 ; 0780
                       D4  A7      9F   00031          PUSHAB  SD_KEYWORDS                                 ;
                   69              02   FB 00034      CALLS   #2, CLI$GET_VALUE                           ;
                       FC  AD      DD   00037          PUSHL   GETCMD_DESC+4                               ; 0781
                       F8  AD      9F   0003A          PUSHAB  GETCMD_DESC                                 ;
            00000000V  EF          02   FB 0003D      CALLS   #2, MAKE_UPPER_CASE                         ;
     08         00             6E   00   2C 00044      MOVC5   #0, (SP), #0, #8, SYSOUTDESC                ; 0785
                     FDE8      C6   00049              ;
                               57   DD 0004C          PUSHL   R7                                          ; 0786
                   68          01   FB 0004E          CALLS   #1, CLI$PRESENT                             ;
                   10          50   E9 00051          BLBC    R0, 1$                                      ;
                FDEB   C6      02   90 00054          MOVB    #2, SYSOUTDESC+3                            ; 0789
                     FDE8      C6   9F   00059        PUSHAB  SYSOUTDESC                                  ; 0790
                               57   DD 0005D          PUSHL   R7                                          ;
                   69          02   FB 0005F          CALLS   #2, CLI$GET_VALUE                           ;
                               0C   11 00062          BRB     2$                                          ;
                FDE8   C6      03   B0 00064  1$:     MOVW    #3, SYSOUTDESC                              ; 0796
                FDEC   C6      C7   9E 00069          MOVAB   P.ABE, SYSOUTDESC+4                         ; 0797
                     00D4      10   A7   9F   00070  2$:  PUSHAB  SD_LIBRARY                               ; 0802
                   68          01   FB 00073          CALLS   #1, CLI$PRESENT                             ;
                   1A          50   E9 00076          BLBC    R0, 3$                                      ;
     08         00             6E   00   2C 00079      MOVC5   #0, (SP), #0, #8, LIBRARYNAME               ; 0805
                       F0      AD   00   0007E         ;
                   F3  AD          02   90 00080       MOVB    #2, LIBRARYNAME+3                           ; 0806
                       F0  AD      9F   00084          PUSHAB  LIBRARYNAME                                 ; 0807
                       10  A7      9F   00087          PUSHAB  SD_LIBRARY                                  ;
                   69              02   FB 0008A      CALLS   #2, CLI$GET_VALUE                           ;
                   52          F0  AD   9E 0008D       MOVAB   LIBRARYNAME, LIBRARY_ADDR                   ; 0808
```

```
                                                02  11 00091          BRB     4$                                      ; 0802
                                                52  D4 00093 3$:       CLRL    LIBRARY_ADDR                            ; 0810
                                                66  D4 00095 4$:       CLRL    HELP_FLAGS                              ; 0816
                                                57  DD 00097          PUSHL   R7                                      ; 0817
                                            68  01  FB 00099          CALLS   #1, CLI$PRESENT
                                  00000000G     8F  50  D1 0009C          CMPL    R0, #CLI$_PRESENT
                                                0E  13 000A3          BEQL    5$
                                                57  DD 000A5          PUSHL   R7                                      ; 0818
                                            68  01  FB 000A7          CALLS   #1, CLI$PRESENT
                                  00000000G     8F  50  D1 000AA          CMPL    R0, #CLI$_NEGATED
                                                14  12 000B1          BNEQ    6$
                                        E4      A7  9F 000B3 5$:       PUSHAB  SD_PROMPT                               ; 0819
                                            68  01  FB 000B6          CALLS   #1, CLI$PRESENT
                                  00000000G     8F  50  D1 000B9          CMPL    R0, #CLI$_DEFAULTED
                                                05  12 000C0          BNEQ    6$
                                            66  01  8A 000C2          BICB2   #1, HELP_FLAGS                          ; 0821
                                                0B  11 000C5          BRB     7$
                                        E4      A7  9F 000C7 6$:       PUSHAB  SD_PROMPT                               ; 0823
                                            68  01  FB 000CA          CALLS   #1, CLI$PRESENT
                  66        01                  00  50  F0 000CD          INSV    R0, #0, #1, HELP_FLAGS
                                            66  0E  8A 000D2 7$:       BICB2   #14, HELP_FLAGS                         ; 0829
                                        48      A7  9F 000D5          PUSHAB  SD_USERLIBRARY                          ; 0830
                                            68  01  FB 000D8          CALLS   #1, CLI$PRESENT
                                            3E  50  E9 000DB          BLBC    R0, 12$
                                        54      A7  9F 000DE          PUSHAB  SD_ALL                                  ; 0833
                                            68  01  FB 000E1          CALLS   #1, CLI$PRESENT
                                            03  50  E9 000E4          BLBC    R0, 8$
                                            66  0E  88 000E7          BISB2   #14, HELP_FLAGS                         ; 0835
                                        60      A7  9F 000EA 8$:       PUSHAB  SD_NONE                                 ; 0836
                                            68  01  FB 000ED          CALLS   #1, CLI$PRESENT
                                            03  50  E9 000F0          BLBC    R0, 9$
                                            66  0E  8A 000F3          BICB2   #14, HELP_FLAGS                         ; 0838
                                          70    A7  9F 000F6 9$:       PUSHAB  SD_PROCESS                              ; 0839
                                            68  01  FB 000F9          CALLS   #1, CLI$PRESENT
                                            03  50  E9 000FC          BLBC    R0, 10$
                                            66  02  88 000FF          BISB2   #2, HELP_FLAGS                          ; 0840
                                        0080    C7  9F 00102 10$:      PUSHAB  SD_GROUP                                ; 0841
                                            68  01  FB 00106          CALLS   #1, CLI$PRESENT
                                            03  50  E9 00109          BLBC    R0, 11$
                                            66  04  88 0010C          BISB2   #4, HELP_FLAGS                          ; 0842
                                        0090    C7  9F 0010F 11$:      PUSHAB  SD_SYSTEM                               ; 0843
                                            68  01  FB 00113          CALLS   #1, CLI$PRESENT
                                            03  50  E9 00116          BLBC    R0, 12$
                                            66  08  88 00119          BISB2   #8, HELP_FLAGS                          ; 0844
                                          F0    A7  9F 0011C 12$:      PUSHAB  SD_PAGE                                 ; 0850
                                            68  01  FB 0011F          CALLS   #1, CLI$PRESENT
      01    A6              01                  00  50  F0 00122          INSV    R0, #0, #1, HELP_FLAGS+1
                                          34    A7  9F 00128          PUSHAB  SD_INSTRUCTIONS                         ; 0851
                                            68  01  FB 0012B          CALLS   #1, CLI$PRESENT
            66              01                  05  50  F0 0012E          INSV    R0, #5, #1, HELP_FLAGS
                                          20    A7  9F 00133          PUSHAB  SD_LIBLIST                              ; 0852
                                            68  01  FB 00136          CALLS   #1, CLI$PRESENT
            66              01                  04  50  F0 00139          INSV    R0, #4, #1, HELP_FLAGS
                                            04  AE  9F 0013E          PUSHAB  SYSOUTNAM                               ; 0857
                                            68  AE  9F 00141          PUSHAB  SYSOUTFAB
                                  00000000V     EF  02  FB 00144          CALLS   #2, OPEN_SYSOUTPUT
                                                5E  DD 0014B          PUSHL   SP                                      ; 0858
```

```
                                    68    AE  9F 0014D          PUSHAB  SYSOUTFAB                                ;
                00000000V           EF        02  FB 00150     CALLS   #2, FIND_FILE_INFO                     ;
                      F8    A6        FC  A6    03  C3 00157     SUBL3   #3, LIST_HEIGHT, CURRENT_HEIGHT      ; 0859
                                  FDF0  C6    B4  0015D          CLRW    OUTPUTDESC                          ; 0860
                          FDF4    C6  FDF8    C6  9E 00161       MOVAB   OUTPUTBUF, OUTPUTDESC+4             ; 0861
                                  04        66  E8 00168       BLBS    HELP_FLAGS, 13$                     ; 0866
                                  07  01    A6  E9 0016B       BLBC    HELP_FLAGS+1, 14$                   ;
                00000000V         EF        00  FB 0016F 13$:  CALLS   #0, OPEN_SYSINPUT                   ; 0867
                                        00000000V EF  9F 00176 14$:  PUSHAB  GET_INPUT                       ; 0872
                                            0044  8F  BB 0017C       PUSHR   #^M<R2,R6>                      ; 0873
                                              F8  AD  9F 00180       PUSHAB  GETCMD_DESC                     ; 0872
                                              10  AE  9F 00183       PUSHAB  LIST_WIDTH                      ;
                                        00000000V EF  9F 00186       PUSHAB  PRINT_HELP_LINE                 ;
                00000000G           00        06  FB 0018C       CALLS   #6, LBR$OUTPUT_HELP                 ;
                                    52        50  D0 00193       MOVL    R0, STATUS                          ;
                                    64    AE  9F 00196       PUSHAB  SYSOUTFAB                           ; 0875
                00000000V           EF        01  FB 00199       CALLS   #1, CLEAN_UP                        ;
                                    50        52  D0 001A0       MOVL    STATUS, R0                          ; 0877
                                              04  001A3       RET                                         ; 0878

; Routine Size:  420 bytes,    Routine Base:  $CODE$ + 0000
```

```
;   292          0879  1 ROUTINE get_input (get_str, prompt_str, out_len) =
;   293          0880  2 BEGIN
;   294          0881  2
;   295          0882  2 !++
;   296          0883  2 ! FUNCTIONAL DESCRIPTION:
;   297          0884  2 !
;   298          0885  2 !       This routine prompts the user and gets a line of text from SYS$INPUT.
;   299          0886  2 !
;   300          0887  2 ! INPUTS:
;   301          0888  2 !
;   302          0889  2 !       get_str =       address of the string descriptor to receive the input string
;   303          0890  2 !
;   304          0891  2 !       prompt_str =    address of the descriptor for the prompt string
;   305          0892  2 !
;   306          0893  2 !       out_len =       address of a longword to receive the length of the input string
;   307          0894  2 !
;   308          0895  2 ! OUTPUTS:
;   309          0896  2 !
;   310          0897  2 !       get_str : as described above
;   311          0898  2 !
;   312          0899  2 ! ROUTINE VALUE:
;   313          0900  2 !
;   314          0901  2 !       The status of the QIO.
;   315          0902  2 !
;   316          0903  2 !--
;   317          0904  2
;   318          0905  2 MAP
;   319          0906  2     get_str : REF $BBLOCK,
;   320          0907  2     prompt_str : REF $BBLOCK;
;   321          0908  2
;   322          0909  2 LOCAL
;   323          0910  2     iosb : VECTOR [2],                                 ! I/O status block
;   324          0911  2     term_char : VECTOR [4],                            ! QIO input termination characters
;   325          0912  2     status;                                            ! Local status
;   326          0913  2
;   327          0914  2 !
;   328          0915  2 ! If paging is in effect, always generate a page break after a prompt.
;   329          0916  2 !
;   330          0917  2 current_height = .list_height - 3;                     ! Generate page break with next help text
;   331          0918  2
;   332          0919  2 !
;   333          0920  2 ! If an answer for this prompt already exists, then skip the prompt and
;   334          0921  2 ! use it.
;   335          0922  2 !
;   336          0923  3 IF (.outputdesc [dsc$w_length] NEQ 0)                   ! If old response is around
;   337          0924  3    THEN BEGIN                                          ! Then use it
;   338          0925  3         get_str [dsc$w_length] = .outputdesc [dsc$w_length];  ! Copy it
;   339          0926  3         CH$MOVE (.outputdesc [dsc$w_length],          !
;   340          0927  3                  .outputdesc [dsc$a_pointer],         !
;   341          0928  3                  .get_str [dsc$a_pointer]);           !
;   342          0929  3         outputdesc [dsc$w_length] = 0;                ! Clear old descriptor
;   343          0930  3         RETURN true;                                  ! Return success
;   344          0931  2         END;                                          !
;   345          0932  2
;   346          0933  2 !
;   347          0934  2 ! If SYS$INPUT is not a terminal, then use LIB$GET_INPUT.  Otherwise,
;   348          0935  2 ! do QIO's to solicit input.
```

```
;  349          0936  2 !
;  350          0937  2 IF .help_flags [hlp$v_notterm]                          ! Is SYS$INPUT a terminal?
;  351          0938  3    THEN IF (status = lib$get_input (.get_str,.prompt_str,.out_len)) ! No, use LIB$GET_INPUT
;  352          0939  3             OR (.status EQL RMS$_EOF)                   !
;  353          0940  2         THEN RETURN .status                             ! Return success or EOF
;  354          0941  2         ELSE SIGNAL_STOP (.status);                     ! Stop if error
;  355          0942  2
;  356          0943  2 !
;  357          0944  2 ! Initialize input termination characters.
;  358          0945  2 ! Use usual set plus '?'.
;  359          0946  2 !
;  360          0947  2 term_char [0] = 8;                                      ! Size of terminator mask
;  361          0948  2 term_char [1] = term_char [2];                          ! Address of terminator mask
;  362          0949  2 term_char [2] = %X'FFFFE0FF';                           ! Quadword terminator mask
;  363          0950  2 term_char [3] = %X'80000000';                           !
;  364          0951  2
;  365          0952  2 !
;  366          0953  2 ! Do QIO and wait for completion.
;  367          0954  2 !
;  368        P 0955  3 IF NOT (status = $QIOW (CHAN = .sysinchan,              ! Input channel
;  369        P 0956  3                         IOSB = iosb,                    ! I/O status block
;  370        P 0957  3                         FUNC = IO$_READVBLK OR IO$_READPROMPT, ! QIO type is read with prompt
;  371        P 0958  3                         P1 = .get_str [dsc$a_pointer],  ! Input buffer
;  372        P 0959  3                         P2 = hlp$c_pagesize,            ! Size of buffer
;  373        P 0960  3                         P4 = term_char,                 ! Terminator mask
;  374        P 0961  3                         P5 = .prompt_str [dsc$a_pointer], ! Prompt string address
;  375          0962  3                         P6 = .prompt_str [dsc$w_length])) ! Prompt string length
;  376          0963  2    THEN SIGNAL_STOP (.status);                          ! Stop if error
;  377          0964  2
;  378          0965  2 get_str [dsc$w_length] = .(iosb[0])<16,16> + 1;         ! Return length of get string
;  379          0966  2
;  380          0967  2 RETURN .status;                                         ! Return QIO status
;  381          0968  1 END;


                                                .EXTRN  SYS$QIOW

                                    00FC 00000 GET_INPUT:
                                                .WORD   Save R2,R3,R4,R5,R6,R7          ; 0879
                          57 00000000G   00  9E 00002   MOVAB   LIB$STOP, R7            ;
                          56 00000000'   EF  9E 00009   MOVAB   OUTPUTDESC, R6          ;
                          5E             18  C2 00010   SUBL2   #24, SP                ;
                0208  C6      020C      C6  03  C3 00013   SUBL3   #3, LIST_HEIGHT, CURRENT_HEIGHT ; 0917
                          51             66  3C 0001B   MOVZWL  OUTPUTDESC, R1         ; 0923
                                         13  13 0001E   BEQL    1$                    ;
                          50       04    AC  D0 00020   MOVL    GET_STR, R0           ; 0925
                          60             51  B0 00024   MOVW    R1, (R0)              ;
                  04  B0       04    B6      51  28 00027   MOVC3   R1, @OUTPUTDESC+4, @4(R0) ; 0928
                                         66  B4 0002D   CLRW    OUTPUTDESC            ; 0929
                          50             01  D0 0002F   MOVL    #1, R0                ; 0930
                                         04  00032   RET                             ;
                      22      0211      C6  06  E1 00033 1$:  BBC     #6, HELP_FLAGS+1, 2$  ; 0937
                          7E       08    AC  7D 00039   MOVQ    PROMPT_STR, -(SP)     ; 0938
                                   04    AC  DD 0003D   PUSHL   GET_STR               ;
                             00000000G   00  FB 00040   CALLS   #3, LIB$GET_INPUT      ;
                          53             50  D0 00047   MOVL    R0, STATUS            ;
```

```
                    62              53  E8 0004A          BLBS    STATUS, 4$                                      ;
          0001827A  8F              53  D1 0004D          CMPL    STATUS, #98938                                  ; 0939
                                    59  13 00054          BEQL    4$                                              ;
                                    53  DD 00056          PUSHL   STATUS                                          ; 0941
                    67              01  FB 00058          CALLS   #1, LIB$STOP                                    ;
                    6E              08  D0 0005B 2$:      MOVL    #8, TERM_CHAR                                   ; 0947
                04  AE        08    AE  9E 0005E          MOVAB   TERM_CHAR+8, TERM_CHAR+4                        ; 0948
                08  AE      EOFF    8F  32 00063          CVTWL   #-7937, TERM_CHAR+8                             ; 0949
                0C  AE  80000000    8F  D0 00069          MOVL    #-2147483648, TERM_CHAR+12                      ; 0950
                    50        08    AC  D0 00071          MOVL    PROMPT_STR, R0                                  ; 0962
                    7E              60  3C 00075          MOVZWL  (R0), -(SP)                                     ;
                              04    A0  DD 00078          PUSHL   4(R0)                                           ;
                              08    AE  9F 0007B          PUSHAB  TERM_CHAR                                       ;
                                    7E  D4 0007E          CLRL    -(SP)                                           ;
                    7E      0200    8F  3C 00080          MOVZWL  #512, -(SP)                                     ;
                    52        04    AC  D0 00085          MOVL    GET_STR, R2                                     ;
                              04    A2  DD 00089          PUSHL   4(R2)                                           ;
                                    7E  7C 0008C          CLRQ    -(SP)                                           ;
                              30    AE  9F 0008E          PUSHAB  IOSB                                            ;
                                    37  DD 00091          PUSHL   #55                                             ;
                              A8    A6  DD 00093          PUSHL   SYSINCHAN                                       ;
                                    7E  D4 00096          CLRL    -(SP)                                           ;
          00000000G 00              0C  FB 00098          CALLS   #12, SYS$QIOW                                   ;
                    53              50  D0 0009F          MOVL    R0, STATUS                                      ;
                    05              53  E8 000A2          BLBS    STATUS, 3$                                      ;
                                    53  DD 000A5          PUSHL   STATUS                                          ; 0963
                    67              01  FB 000A7          CALLS   #1, LIB$STOP                                    ;
              62        12    AE      01  A1 000AA 3$:     ADDW3   #1, IOSB+2, (R2)                                ; 0965
                    50              53  D0 000AF 4$:      MOVL    STATUS, R0                                      ; 0967
                                        04 000B2          RET                                                     ; 0968

; Routine Size:  179 bytes,    Routine Base:  $CODE$ + 01A4
```

```
;   383          0969  1 ROUTINE print_help_line (linedesc) =
;   384          0970  2 BEGIN
;   385          0971  2
;   386          0972  2 !++
;   387          0973  2 ! FUNCTIONAL DESCRIPTION:
;   388          0974  2 !
;   389          0975  2 !       Driver for the two output routines in this module.
;   390          0976  2 !       Calls put_page_brk and put_output.
;   391          0977  2 !
;   392          0978  2 ! INPUTS:
;   393          0979  2 !
;   394          0980  2 !       linedesc =      address of string descriptor for the line of help
;   395          0981  2 !                       text to be output
;   396          0982  2 !
;   397          0983  2 ! OUTPUTS:
;   398          0984  2 !
;   399          0985  2 !       None.
;   400          0986  2 !
;   401          0987  2 ! ROUTINE VALUE:
;   402          0988  2 !
;   403          0989  2 !       Status returned by put_page_break
;   404          0990  2 !
;   405          0991  2 !--
;   406          0992  2
;   407          0993  2 IF NOT put_page_break ()                                ! Put a page break
;   408          0994  2    THEN RETURN LBR$_ENDTOPIC;                           ! Then terminate listing
;   409          0995  2 put_output (.linedesc);                                 ! Output help to terminal
;   410          0996  2
;   411          0997  2 RETURN true;
;   412          0998  1 END;


                                0000 00000 PRINT_HELP_LINE:
                                           .WORD   Save nothing                 ;  0969
00000000V  EF                   00  FB 00002           CALLS   #0, PUT_PAGE_BREAK           ;  0993
           08                   50  E8 00009           BLBS    R0, 1$                       ;
           50 00000000G         8F  D0 0000C           MOVL    #LBR$_ENDTOPIC, R0           ;  0994
                                04  00013           RET                                  ;
                      04        AC  DD 00014 1$:       PUSHL   LINEDESC                     ;  0995
00000000V  EF                   01  FB 00017           CALLS   #1, PUT_OUTPUT               ;
           50                   01  D0 0001E           MOVL    #1, R0                       ;  0997
                                04  00021           RET                                  ;  0998

; Routine Size:  34 bytes,    Routine Base:  $CODE$ + 0257
```

```
;  414     0999  1 ROUTINE put_page_break =
;  415     1000  2 BEGIN
;  416     1001  2
;  417     1002  2 !++
;  418     1003  2 ! FUNCTIONAL DESCRIPTION:
;  419     1004  2 !
;  420     1005  2 !       If output is going to a video terminal, and paging is enabled,
;  421     1006  2 !       then a page break is forced if the screen is full of help text.
;  422     1007  2 !
;  423     1008  2 ! INPUTS:
;  424     1009  2 !
;  425     1010  2 !       None.
;  426     1011  2 !
;  427     1012  2 ! OUTPUTS:
;  428     1013  2 !
;  429     1014  2 !       None.
;  430     1015  2 !
;  431     1016  2 ! ROUTINE VALUE:
;  432     1017  2 !
;  433     1018  2 !       False, if user responds with a '?' to a conditional page break.
;  434     1019  2 !       True, otherwise.
;  435     1020  2 !
;  436     1021  2 !--
;  437     1022  2
;  438     1023  2 LOCAL
;  439     1024  2     status,                                         ! Local status longword
;  440     1025  2     prompt_desc : $BBLOCK [dsc$c_s_bln];            ! Descriptor for page break prompt
;  441     1026  2
;  442     1027  2 IF .help_flags [hlp$v_page]                         ! If paging enabled
;  443     1028  3 THEN IF (.current_height LSS .list_height - 4)      ! Then if not three or less lines left on th
;  444     1029  2      THEN current_height = .current_height + 1      ! Then simply increment the line count
;  445     1030  3      ELSE BEGIN                                     ! Else clear the screen
;  446     1031  4           IF (.current_height EQL .list_height - 4) ! And if in the middle of some help informat
;  447     1032  4           THEN BEGIN                                ! Then output a page break prompt
;  448     1033  4
;  449     1034  4                outputdesc [dsc$w_length] = 1;       ! Init blank string
;  450     1035  4                outputbuf [0,0,8,0] = 10;            !
;  451     1036  4                put_output (outputdesc);             ! Output a blank line
;  452     1037  4
;  453     1038  4                outputdesc [dsc$w_length] = 0;       ! Init input string
;  454     1039  4                status = get_input (outputdesc,      ! Issue page break prompt
;  455     1040  4                          pagebrk_prompt, outputdesc);!
;  456     1041  4                IF .status EQL RMS$_EOF              ! If EOF was detected
;  457     1042  4                   THEN $EXIT();                     ! Then exit help now
;  458     1043  4
;  459     1044  5                SELECTONE (CH$RCHAR (.outputdesc[dsc$a_pointer] ! Test termination character
;  460     1045  4                   + .outputdesc [dsc$w_length] - 1)) OF SET    !
;  461     1046  4
;  462     1047  4                [%X'1A']:                            ! If CTRL/Z
;  463     1048  4                $EXIT();                             ! Then exit help now
;  464     1049  4
;  465     1050  4                [%C'?']:                             ! If ?
;  466     1051  4                RETURN false;                        ! Simply continue
;  467     1052  4
;  468     1053  4                [OTHERWISE]:                         ! Anything else
;  469     1054  4                IF .outputdesc [dsc$w_length] EQL 1  ! If no text
;  470     1055  4                THEN outputdesc[dsc$w_length] = 0    ! Then simply continue
```

```
;  471     1056  5                 ELSE BEGIN                                          ! Use text as next command
;  472     1057  5                      outputdesc [dsc$w_length] =                    !
;  473     1058  5                         .outputdesc [dsc$w_length] - 1;             !
;  474     1059  5                      RETURN false;                                  !
;  475     1060  4                      END;                                           !
;  476     1061  4
;  477     1062  4                 TES;
;  478     1063  3                 END;
;  479     1064  3
;  480     1065  3             scr$erase_page (1,1);                                   ! Clear screen
;  481     1066  2             current_height = 1;                                     ! Reset line count
;  482     1067  2             END;                                                    !
;  483     1068  2
;  484     1069  2 RETURN true;
;  485     1070  1 END;


                                                        .EXTRN  SYS$EXIT

                                        000C 00000 PUT_PAGE_BREAK:
                                                        .WORD   Save R2,R3                      ; 0999
                          53 00000000G  00   9E 00002   MOVAB   SYS$EXIT, R3                    ;
                          52 00000000'  EF   9E 00009   MOVAB   OUTPUTDESC, R2                  ;
                          5E            08   C2 00010   SUBL2   #8, SP                        ;
                          7D      0211  C2   E9 00013   BLBC    HELP_FLAGS+1, 6$                ; 1027
                      50      020C  C2   04   C3 00018   SUBL3   #4, LIST_HEIGHT, R0          ; 1028
                          50      0208  C2   D1 0001E   CMPL    CURRENT_HEIGHT, R0              ;
                                        06   18 00023   BGEQ    1$                              ;
                                  0208  C2   D6 00025   INCL    CURRENT_HEIGHT                  ; 1029
                                        6A   11 00029   BRB     6$                              ;
                                        58   12 0002B 1$: BNEQ    5$                              ; 1031
                              62        01   B0 0002D   MOVW    #1, OUTPUTDESC                  ; 1034
                          08  A2        0A   90 00030   MOVB    #10, OUTPUTBUF                  ; 1035
                                        52   DD 00034   PUSHL   R2                              ; 1036
                  00000000V   EF        01   FB 00036   CALLS   #1, PUT_OUTPUT                 ;
                                        62   B4 0003D   CLRW    OUTPUTDESC                      ; 1038
                                        52   DD 0003F   PUSHL   R2                              ; 1039
                                00000000' EF   9F 00041   PUSHAB  PAGEBRK_PROMPT                 ;
                                        52   DD 00047   PUSHL   R2                              ;
                            FEDD  CF        03   FB 00049   CALLS   #3, GET_INPUT               ;
                      0001827A  8F        50   D1 0004E   CMPL    STATUS, #98938                  ; 1041
                                        05   12 00055   BNEQ    2$                              ;
                                        01   DD 00057   PUSHL   #1                              ; 1042
                              63        01   FB 00059   CALLS   #1, SYS$EXIT                    ;
                              50        62   3C 0005C 2$: MOVZWL  OUTPUTDESC, R0               ; 1045
                              50  04    A2   C0 0005F   ADDL2   OUTPUTDESC+4, R0                ;
                              50  FF    A0   9A 00063   MOVZBL  -1(R0), R0                      ;
                              1A        50   91 00067   CMPB    R0, #26                         ; 1047
                                        07   12 0006A   BNEQ    3$                              ;
                                        01   DD 0006C   PUSHL   #1                              ; 1048
                              63        01   FB 0006E   CALLS   #1, SYS$EXIT                    ;
                                        12   11 00071   BRB     5$                              ;
                              3F        50   91 00073 3$: CMPB    R0, #63                         ; 1050
                                        21   13 00076   BEQL    7$                              ;
                              01        62   B1 00078   CMPW    OUTPUTDESC, #1                  ; 1054
                                        04   12 0007B   BNEQ    4$                              ;
```

```
                                62  B4 0007D          CLRW    OUTPUTDESC                              ; 1055
                                04  11 0007F          BRB     5$                                      ;
                                62  B7 00081 4$:      DECW    OUTPUTDESC                              ; 1058
                                14  11 00083          BRB     7$                                      ; 1059
                                01  DD 00085 5$:      PUSHL   #1                                      ; 1065
                                01  DD 00087          PUSHL   #1                                      ;
                    00000000G   00  02  FB 00089      CALLS   #2, SCR$ERASE_PAGE                      ;
                        0208    C2  01  D0 00090      MOVL    #1, CURRENT_HEIGHT                      ; 1066
                                50  01  D0 00095 6$:      MOVL    #1, R0                              ; 1069
                                        04 00098      RET                                             ;
                                50  D4 00099 7$:      CLRL    R0                                      ; 1070
                                        04 0009B      RET                                             ;

; Routine Size:  156 bytes,    Routine Base:  $CODE$ + 0279
```

```
;  487          1071  1 ROUTINE put_output (linedesc) =
;  488          1072  2 BEGIN
;  489          1073  2
;  490          1074  2 !++
;  491          1075  2 ! FUNCTIONAL DESCRIPTION:
;  492          1076  2 !
;  493          1077  2 !       Put a line of help text to SYS$OUTPUT.
;  494          1078  2 !
;  495          1079  2 ! INPUTS:
;  496          1080  2 !
;  497          1081  2 !       linedesc =      address of string descriptor for the line of help
;  498          1082  2 !                       text to be output
;  499          1083  2 !
;  500          1084  2 ! OUTPUTS:
;  501          1085  2 !
;  502          1086  2 !       None.
;  503          1087  2 !
;  504          1088  2 ! ROUTINE VALUE:
;  505          1089  2 !
;  506          1090  2 !       Always true.
;  507          1091  2 !
;  508          1092  2 !--
;  509          1093  2
;  510          1094  2 MAP
;  511          1095  2     linedesc : REF $BBLOCK;
;  512          1096  2
;  513          1097  2 LOCAL
;  514          1098  2     linebuf : $BBLOCK [hlp$c_pagesize],
;  515          1099  2     status;
;  516          1100  2
;  517          1101  2 sysoutrab [rab$w_rsz] = 0;
;  518          1102  2 sysoutrab [rab$l_rbf] = linebuf;
;  519          1103  2
;  520          1104  2 IF .linedesc [dsc$w_length] NEQ 0
;  521          1105  3    THEN BEGIN
;  522          1106  3         CH$MOVE (.linedesc [dsc$w_length], .linedesc [dsc$a_pointer], linebuf);
;  523          1107  3         sysoutrab [rab$w_rsz] = .linedesc [dsc$w_length];
;  524          1108  2         END;
;  525          1109  2
;  526          1110  3 IF NOT (status = $PUT (RAB = sysoutrab))
;  527          1111  2    THEN SIGNAL_STOP ( (shr$_writeerr OR hlp$c_facility OR sts$k_error),
;  528          1112  2         1, sysout_name, .status, .sysoutrab [rab$l_stv]);
;  529          1113  2
;  530          1114  2 RETURN true;
;  531          1115  1 END;                                                    !Of put_output


                                                .EXTRN  SYS$PUT

                                00FC 00000 PUT_OUTPUT:
                                                .WORD   Save R2,R3,R4,R5,R6,R7          ; 1071
                     57 00000000'   EF  9E 00002        MOVAB   SYSOUTRAB+34, R7        ;
                     5E      FE00   CE  9E 00009        MOVAB   -512(SP), SP            ;
                                    67  B4 0000E        CLRW    SYSOUTRAB+34            ; 1101
                  06 A7             6E  9E 00010        MOVAB   LINEBUF, SYSOUTRAB+40   ; 1102
                     56       04    AC  D0 00014        MOVL    LINEDESC, R6            ; 1104
```

```
                                        66  B5 00018          TSTW    (R6)                                                    ;
                                        08  13 0001A          BEQL    1$                                                      ;
                    6E      04  B6      66  28 0001C          MOVC3   (R6), @4(R6), LINEBUF                                   ; 1106
                                67      66  B0 00021          MOVW    (R6), SYSOUTRAB+34                                      ; 1107
                                    DE  A7  9F 00024 1$:       PUSHAB  SYSOUTRAB                                               ; 1110
                00000000G       00      01  FB 00027          CALLS   #1, SYS$PUT                                             ;
                                17      50  E8 0002E          BLBS    STATUS, 2$                                              ;
                                    EA  A7  DD 00031          PUSHL   SYSOUTRAB+12                                            ; 1112
                                        50  DD 00034          PUSHL   STATUS                                                  ;
                                    22  A7  9F 00036          PUSHAB  SYSOUT_NAME                                             ; 1111
                                        01  DD 00039          PUSHL   #1                                                      ;
                                007610D2    8F  DD 0003B      PUSHL   #7737554                                                ;
                00000000G       00      05  FB 00041          CALLS   #5, LIB$STOP                                            ;
                                50      01  D0 00048 2$:      MOVL    #1, R0                                                  ; 1114
                                        04  0004B             RET                                                             ; 1115

; Routine Size:  76 bytes,    Routine Base:  $CODE$ + 0315
```

```
;  533          1116  1 ROUTINE find_file_info (fab, listwidth) =
;  534          1117  2 BEGIN
;  535          1118  2
;  536          1119  2 !++
;  537          1120  2 ! FUNCTIONAL DESCRIPTION:
;  538          1121  2 !
;  539          1122  2 !       Determine the file characteristics, i.e., page height and line width,
;  540          1123  2 !       of the file specified by the input fab.  Also, check to see if device
;  541          1124  2 !       is suitable for page breaks.
;  542          1125  2 !
;  543          1126  2 ! INPUTS:
;  544          1127  2 !
;  545          1128  2 !       fab =           address of FAB for file of interest
;  546          1129  2 !
;  547          1130  2 !       listwidth =     address of longword to contain line width
;  548          1131  2 !
;  549          1132  2 ! OUTPUTS:
;  550          1133  2 !
;  551          1134  2 !       listwidth : as described above.
;  552          1135  2 !       Also implicitly:  list_height and help_flags.
;  553          1136  2 !
;  554          1137  2 ! ROUTINE VALUE:
;  555          1138  2 !
;  556          1139  2 !       Always true.
;  557          1140  2 !
;  558          1141  2 !--
;  559          1142  2
;  560          1143  2 MAP
;  561          1144  2     fab : REF $BBLOCK;
;  562          1145  2
;  563          1146  2 BIND
;  564          1147  2     namblk = .fab [fab$l_nam] : $BBLOCK;        ! NAM block associated with input FAB
;  565          1148  2
;  566          1149  2 MACRO
;  567          1150  2     ddp$b_pagelen = 3,0,8,0%;                  ! *** Hardwired page length offset ***
;  568          1151  2
;  569          1152  2 LITERAL
;  570          1153  2     getdvilen = 4*12 + 4;
;  571          1154  2
;  572          1155  2 LOCAL
;  573          1156  2     getdvidesc : $BBLOCK [getdvilen],
;  574          1157  2     devnamdesc : $BBLOCK [dsc$c_s_bln],
;  575          1158  2     devbufsiz,
;  576          1159  2     devclass,
;  577          1160  2     devdepend : $BBLOCK [4],
;  578          1161  2     devtype;
;  579          1162  2
;  580          1163  2 .listwidth = hlp$c_liswidth;                   ! Assume default width of 80
;  581          1164  2
;  582          1165  2 devnamdesc [dsc$w_length] = .namblk [nam$b_dev];   ! Get the device name
;  583          1166  2 devnamdesc [dsc$a_pointer] = .namblk [nam$l_dev];  !
;  584          1167  2
;  585          1168  2 !
;  586          1169  2 ! Do a $GETDVI to get the parameters of interest.
;  587          1170  2 !
;  588          1171  2 CH$FILL (0, getdvilen, getdvidesc);             ! Init the $GETDVI buffer
;  589          1172  2
```

```
;  590     1173  2 getdvidesc [0,0,16,0] = 4;                                  ! Get the device class
;  591     1174  2 getdvidesc [2,0,16,0] = dvi$_devclass;                      !
;  592     1175  2 getdvidesc [4,0,32,0] = devclass;                           !
;  593     1176  2 getdvidesc [8,0,32,0] = getdvidesc [0,0,16,0];              !
;  594     1177  2
;  595     1178  2 getdvidesc [12,0,16,0] = 4;                                 ! Get the device dependent chars
;  596     1179  2 getdvidesc [14,0,16,0] = dvi$_devdepend;                     !
;  597     1180  2 getdvidesc [16,0,32,0] = devdepend;                         !
;  598     1181  2 getdvidesc [20,0,32,0] = getdvidesc [12,0,16,0];            !
;  599     1182  2
;  600     1183  2 getdvidesc [24,0,16,0] = 4;                                 ! Get the device type
;  601     1184  2 getdvidesc [26,0,16,0] = dvi$_devtype;                      !
;  602     1185  2 getdvidesc [28,0,32,0] = devtype;                           !
;  603     1186  2 getdvidesc [32,0,32,0] = getdvidesc [24,0,16,0];            !
;  604     1187  2
;  605     1188  2 getdvidesc [36,0,16,0] = 4;                                 ! Get the device buffer size
;  606     1189  2 getdvidesc [38,0,16,0] = dvi$_devbufsiz;                    !
;  607     1190  2 getdvidesc [40,0,32,0] = devbufsiz;                         !
;  608     1191  2 getdvidesc [44,0,32,0] = getdvidesc [36,0,16,0];            !
;  609     1192  2
;  610     1193  2 IF (.fab [fab$l_dev] AND dev$m_spl) NEQ 0                   ! If output device is spooled
;  611     1194  3    THEN BEGIN                                               ! Then use the secondary device
;  612     1195  3         getdvidesc [2,0,16,0] = dvi$_devclass OR dvi$c_secondary;
;  613     1196  3         getdvidesc [14,0,16,0] = dvi$_devdepend OR dvi$c_secondary;
;  614     1197  3         getdvidesc [26,0,16,0] = dvi$_devtype OR dvi$c_secondary;
;  615     1198  3         getdvidesc [38,0,16,0] = dvi$_devbufsiz OR dvi$c_secondary;
;  616     1199  2         END;
;  617     1200  2
;  618     1201  3 IF $GETDVI (DEVNAM = devnamdesc, ITMLST = getdvidesc)       ! Get the device characteristics
;  619     1202  3    THEN BEGIN                                               !
;  620     1203  3         .listwidth = MINU (.devbufsiz, hlp$c_maxwidth);     ! Get the listing width
;  621     1204  3         list_height = .devdepend [ddp$b_pagelen];           ! Get the listing height
;  622     1205  4         IF (.devclass NEQ dc$_term) OR                      ! If output device is not a terminal
;  623     1206  4                (NOT .devdepend [tt$v_scope]) OR             ! Or output device is not a video terminal
;  624     1207  4                (.devtype EQL dt$_ttyunkn))                  ! Or terminal type is unknown
;  625     1208  3            THEN help_flags = .help_flags AND NOT hlp$m_page;! Then do not generate page breaks
;  626     1209  2         END;                                                !
;  627     1210  2
;  628     1211  2 RETURN true;
;  629     1212  1 END;                                                       !Of find_file_info
; INFO#250                  L1:1203
; Referenced LOCAL symbol DEVBUFSIZ is probably not initialized
; INFO#250                  L1:1205
; Referenced LOCAL symbol DEVCLASS is probably not initialized
; INFO#250                  L1:1207
; Referenced LOCAL symbol DEVTYPE is probably not initialized


                                                .EXTRN  SYS$GETDVI

                                    007C 00000 FIND_FILE_INFO:
                                                .WORD   Save R2,R3,R4,R5,R6     ; 1116
                                5E      B4  AE  9E 00002        MOVAB   -76(SP), SP     ;
                                56      04  AC  D0 00006        MOVL    FAB, R6         ; 1147
                                50      28  A6  D0 0000A        MOVL    40(R6), R0      ;
                            08  BC      50  8F  9A 0000E        MOVZBL  #80, @LISTWIDTH ; 1163
```

```
                        10   AE      39   A0  9B 00013          MOVZBW  57(R0), DEVNAMDESC                        ; 1165
                        14   AE      44   A0  D0 00018          MOVL    68(R0), DEVNAMDESC+4                     ; 1166
              34        00           6E        00  2C 0001D    MOVC5   #0, (SP), #0, #52, GETDVIDESC            ; 1171
                                        18   AE  00022
                        18   AE 00040004   8F  D0 00024          MOVL    #262148, GETDVIDESC                      ; 1173
                        1C   AE           6E  9E 0002C          MOVAB   DEVCLASS, GETDVIDESC+4                   ; 1175
                        20   AE        18   AE  9E 00030          MOVAB   GETDVIDESC, GETDVIDESC+8                 ; 1176
                        24   AE 000A0004   8F  D0 00035          MOVL    #655364, GETDVIDESC+12                   ; 1178
                        28   AE        0C   AE  9E 0003D          MOVAB   DEVDEPEND, GETDVIDESC+16                 ; 1180
                        2C   AE        24   AE  9E 00042          MOVAB   GETDVIDESC+12, GETDVIDESC+20             ; 1181
                        30   AE 00060004   8F  D0 00047          MOVL    #393220, GETDVIDESC+24                   ; 1183
                        34   AE        04   AE  9E 0004F          MOVAB   DEVTYPE, GETDVIDESC+28                   ; 1185
                        38   AE        30   AE  9E 00054          MOVAB   GETDVIDESC+24, GETDVIDESC+32             ; 1186
                        3C   AE 00080004   8F  D0 00059          MOVL    #524292, GETDVIDESC+36                   ; 1188
                        40   AE        08   AE  9E 00061          MOVAB   DEVBUFSIZ, GETDVIDESC+40                 ; 1190
                        44   AE        3C   AE  9E 00066          MOVAB   GETDVIDESC+36, GETDVIDESC+44             ; 1191
              10        40   A6           06  E1 0006B          BBC     #6, 64(R6), 1$                           ; 1193
                        1A   AE           05  B0 00070          MOVW    #5, GETDVIDESC+2                         ; 1195
                        26   AE           0B  B0 00074          MOVW    #11, GETDVIDESC+14                       ; 1196
                        32   AE           07  B0 00078          MOVW    #7, GETDVIDESC+26                        ; 1197
                        3E   AE           09  B0 0007C          MOVW    #9, GETDVIDESC+38                        ; 1198
                                          7E  7C 00080 1$:      CLRQ    -(SP)                                    ; 1201
                                          7E  7C 00082          CLRQ    -(SP)
                                    28    AE  9F 00084          PUSHAB  GETDVIDESC
                                    24    AE  9F 00087          PUSHAB  DEVNAMDESC
                                          7E  7C 0008A          CLRQ    -(SP)
                00000000G   00              08  FB 0008C          CALLS   #8, SYS$GETDVI
                            37              50  E9 00093          BLBC    R0, 4$
                            50        08    AE  D0 00096          MOVL    DEVBUFSIZ, R0                            ; 1203
                00000084    8F              50  D1 0009A          CMPL    R0, #132
                                            04  1B 000A1          BLEQU   2$
                            50        84    8F  9A 000A3          MOVZBL  #132, R0
                                  08    BC  50  D0 000A7 2$:      MOVL    R0, @LISTWIDTH
                00000000'   EF        0F    AE  9A 000AB          MOVZBL  DEVDEPEND+3, LIST_HEIGHT                 ; 1204
                00000042    8F              6E  D1 000B3          CMPL    DEVCLASS, #66                            ; 1205
                                            0A  12 000BA          BNEQ    3$
                      05        0D    AE      04  E1 000BC          BBC     #4, DEVDEPEND+1, 3$                      ; 1206
                                        04  AE  D5 000C1          TSTL    DEVTYPE                                  ; 1207
                                            07  12 000C4          BNEQ    4$
                00000000'   EF              01  8A 000C6 3$:      BICB2   #1, HELP_FLAGS+1                        ; 1208
                            50              01  D0 000CD 4$:      MOVL    #1, R0                                   ; 1211
                                                04 000D0          RET                                              ; 1212

; Routine Size:  209 bytes,    Routine Base:  $CODE$ + 0361
```

```
;  631          1213  1 ROUTINE open_sysinput =
;  632          1214  2 BEGIN
;  633          1215  2
;  634          1216  2 !++
;  635          1217  2 ! FUNCTIONAL DESCRIPTION:
;  636          1218  2 !
;  637          1219  2 !       Open SYS$INPUT.
;  638          1220  2 !
;  639          1221  2 ! INPUTS:
;  640          1222  2 !
;  641          1223  2 !       None.
;  642          1224  2 !
;  643          1225  2 ! OUTPUTS:
;  644          1226  2 !
;  645          1227  2 !       sysinchan =     longword containing channel assigned to SYS$INPUT
;  646          1228  2 !
;  647          1229  2 ! ROUTINE VALUE:
;  648          1230  2 !
;  649          1231  2 !       Always true.
;  650          1232  2 !
;  651          1233  2 !--
;  652          1234  2
;  653          1235  2 LOCAL
;  654          1236  2     devinfobuf : $BBLOCK [dib$k_length],                ! DIB buffer
;  655          1237  2     devinfodesc : $BBLOCK [dsc$c_s_bln],                ! DIB desc
;  656          1238  2     sysinstring1 : VECTOR [nam$c_maxrss, BYTE],         ! Space for SYS$INPUT resultant string
;  657          1239  2     sysinstring2 : VECTOR [nam$c_maxrss, BYTE],         ! Space for SYS$INPUT resultant string
;  658          1240  2     sysindesc : $BBLOCK [dsc$c_s_bln],                  ! String descriptor for SYS$INPUT
;  659          1241  2     sysinname : $BBLOCK [dsc$c_s_bln],                  ! String descriptor for resultant name
;  660          1242  2     status;
;  661          1243  2
;  662          1244  2 sysindesc [dsc$w_length] = .sysinput [dsc$w_length];    ! Init input desc
;  663          1245  2 sysindesc [dsc$a_pointer] = .sysinput [dsc$a_pointer];
;  664          1246  2
;  665          1247  2 sysinname [dsc$w_length] = nam$c_maxrss;                ! Init output desc
;  666          1248  2 sysinname [dsc$a_pointer] = sysinstring2;
;  667          1249  2
;  668        P 1250  3 WHILE (status = $TRNLOG (LOGNAM = sysindesc,           ! Recursively translate input desc
;  669        P 1251  3                          RSLLEN = sysinname [dsc$w_length],
;  670          1252  3                          RSLBUF = sysinname))
;  671          1253  3 DO BEGIN
;  672          1254  3
;  673          1255  4     IF (.status EQL SS$_NOTRAN)                        ! Stop when not translatable
;  674          1256  3        THEN EXITLOOP
;  675          1257  3        ELSE IF NOT .status                            ! Signal any errors
;  676          1258  3                THEN SIGNAL_STOP (.status);
;  677          1259  3
;  678          1260  3     IF CH$RCHAR (.sysinname [dsc$a_pointer]) EQL %X'1B'
;  679          1261  4        THEN BEGIN
;  680          1262  4             sysinname [dsc$w_length] = .sysinname [dsc$w_length] - 4;
;  681          1263  4             sysinname [dsc$a_pointer] = .sysinname [dsc$a_pointer] + 4;
;  682          1264  3             END;
;  683          1265  3     sysindesc [dsc$w_length] = .sysinname [dsc$w_length];  ! New input desc
;  684          1266  3     sysindesc [dsc$a_pointer] = .sysinname [dsc$a_pointer];
;  685          1267  3     sysinname [dsc$w_length] = nam$c_maxrss;               ! New output desc
;  686          1268  3     sysinname [dsc$a_pointer] =
;  687          1269  4          (IF .sysinname [dsc$a_pointer] EQL sysinstring1
```

```
;  688          1270  4              THEN sysinstring2
;  689          1271  3              ELSE sysinstring1);
;  690          1272  2     END;
;  691          1273  2
;  692          1274  3 IF NOT (status = $ASSIGN (DEVNAM = sysinname, CHAN = sysinchan))
;  693          1275  2     THEN SIGNAL_STOP (.status);
;  694          1276  2
;  695          1277  2 devinfodesc [dsc$w_length] = dib$k_length;
;  696          1278  2 devinfodesc [dsc$a_pointer] = devinfobuf;
;  697          1279  3 IF $GETCHN (CHAN = .sysinchan, SCDBUF = devinfodesc)
;  698          1280  3     THEN IF (.devinfobuf [dib$b_devclass] NEQ dc$_term)
;  699          1281  3          THEN BEGIN
;  700          1282  3               $DASSGN (CHAN = .sysinchan);
;  701          1283  3               help_flags [hlp$v_notterm] = true;
;  702          1284  2               END;
;  703          1285  2
;  704          1286  2 RETURN true;
;  705          1287  1 END;


                                                .EXTRN  SYS$TRNLOG, SYS$ASSIGN
                                                .EXTRN  SYS$GETCHN, SYS$DASSGN

                                001C 00000 OPEN_SYSINPUT:
                                                .WORD   Save R2,R3,R4                   ; 1213
                         54 00000000G   00   9E 00002           MOVAB   LIB$STOP, R4            ;
                         53 00000000'   EF   9E 00009           MOVAB   SYSINCHAN, R3           ;
                         5E       FD74  CE   9E 00010           MOVAB   -652(SP), SP            ;
                      08 AE 00000000'   EF   B0 00015           MOVW    SYSINPUT, SYSINDESC     ; 1244
                      0C AE 00000000'   EF   D0 0001D           MOVL    SYSINPUT+4, SYSINDESC+4 ; 1245
                         6E         FF   8F   9B 00025           MOVZBW  #255, SYSINNAME         ; 1247
                      04 AE         10   AE   9E 00029           MOVAB   SYSINSTRING2, SYSINNAME+4 ; 1248
                                         7E   7C 0002E 1$:      CLRQ    -(SP)                   ; 1252
                                         7E   D4 00030          CLRL    -(SP)                   ;
                                      0C AE   9F 00032          PUSHAB  SYSINNAME               ;
                                      10 AE   9F 00035          PUSHAB  SYSINNAME               ;
                                      1C AE   9F 00038          PUSHAB  SYSINDESC               ;
                   00000000G          00      06 FB 0003B       CALLS   #6, SYS$TRNLOG          ;
                                      52      50 D0 00042       MOVL    R0, STATUS              ;
                                      47      52 E9 00045       BLBC    STATUS, 6$              ;
                   00000629           8F      52 D1 00048       CMPL    STATUS, #1577           ; 1255
                                              3E 13 0004F       BEQL    6$                      ;
                                      05      52 E8 00051       BLBS    STATUS, 2$              ; 1257
                                              52 DD 00054       PUSHL   STATUS                  ; 1258
                                      64      01 FB 00056       CALLS   #1, LIB$STOP            ;
                                      1B   04 BE 91 00059 2$:   CMPB    @SYSINNAME+4, #27       ; 1260
                                              07 12 0005D       BNEQ    3$                      ;
                                      6E      04 A2 0005F       SUBW2   #4, SYSINNAME           ; 1262
                                   04 AE      04 C0 00062       ADDL2   #4, SYSINNAME+4         ; 1263
                                   08 AE      6E B0 00066 3$:   MOVW    SYSINNAME, SYSINDESC    ; 1265
                                   0C AE   04 AE D0 0006A       MOVL    SYSINNAME+4, SYSINDESC+4 ; 1266
                                      6E   FF 8F 9B 0006F       MOVZBW  #255, SYSINNAME         ; 1267
                                      50 0110 CE 9E 00073       MOVAB   SYSINSTRING1, R0        ; 1269
                                      50   04 AE D1 00078       CMPL    SYSINNAME+4, R0         ;
                                              06 12 0007C       BNEQ    4$                      ;
                                      50   10 AE 9E 0007E       MOVAB   SYSINSTRING2, R0        ;
```

```
                                05  11 00082       BRB     5$                            ;
          50      0110          CE  9E 00084 4$:   MOVAB   SYSINSTRING1, R0              ;
    04    AE                    50  D0 00089 5$:   MOVL    R0, SYSINNAME+4               ;
                                9F  11 0008D       BRB     1$                            ; 1250
                                7E  7C 0008F 6$:   CLRQ    -(SP)                         ; 1274
                                53  DD 00091       PUSHL   R3                            ;
                    0C          AE  9F 00093       PUSHAB  SYSINNAME                     ;
     00000000G    00            04  FB 00096       CALLS   #4, SYS$ASSIGN                ;
                  52            50  D0 0009D       MOVL    R0, STATUS                    ;
                  05            52  E8 000A0       BLBS    STATUS, 7$                    ;
                                52  DD 000A3       PUSHL   STATUS                        ; 1275
                  64            01  FB 000A5       CALLS   #1, LIB$STOP                  ;
    84    AD        74          8F  9B 000A8 7$:   MOVZBW  #116, DEVINFODESC             ; 1277
    88    AD        8C          AD  9E 000AD       MOVAB   DEVINFOBUF, DEVINFODESC+4     ; 1278
                    84          AD  9F 000B2       PUSHAB  DEVINFODESC                   ; 1279
                                7E  7C 000B5       CLRQ    -(SP)                         ;
                                7E  D4 000B7       CLRL    -(SP)                         ;
                                63  DD 000B9       PUSHL   SYSINCHAN                     ;
     00000000G    00            05  FB 000BB       CALLS   #5, SYS$GETCHN                ;
                  16            50  E9 000C2       BLBC    R0, 8$                        ;
    42    8F        90          AD  91 000C5       CMPB    DEVINFOBUF+4, #66             ; 1280
                                0F  13 000CA       BEQL    8$                            ;
                                63  DD 000CC       PUSHL   SYSINCHAN                     ; 1282
     00000000G    00            01  FB 000CE       CALLS   #1, SYS$DASSGN                ;
        0269      C3        40  8F  88 000D5       BISB2   #64, HELP_FLAGS+1             ; 1283
                  50            01  D0 000DB 8$:   MOVL    #1, R0                        ; 1286
                                    04 000DE       RET                                   ; 1287

; Routine Size:  223 bytes,    Routine Base:  $CODE$ + 0432
```

```
;   707          1288  1 ROUTINE open_sysoutput (sysoutfab, sysoutnam)  =
;   708          1289  2 BEGIN
;   709          1290  2
;   710          1291  2 !++
;   711          1292  2 ! FUNCTIONAL DESCRIPTION:
;   712          1293  2 !
;   713          1294  2 !       Open SYS$OUTPUT.
;   714          1295  2 !
;   715          1296  2 ! INPUTS:
;   716          1297  2 !
;   717          1298  2 !       sysoutfab =     address of FAB for SYS$OUTPUT
;   718          1299  2 !
;   719          1300  2 !       sysoutnam =     address of NAM block for SYS$OUTPUT
;   720          1301  2 !
;   721          1302  2 ! OUTPUTS:
;   722          1303  2 !
;   723          1304  2 !       sysoutfab, sysoutrab, sysout_name : updated as expected.
;   724          1305  2 !
;   725          1306  2 ! ROUTINE VALUE:
;   726          1307  2 !
;   727          1308  2 !       Always true.
;   728          1309  2 !
;   729          1310  2 !--
;   730          1311  2
;   731          1312  2 MAP
;   732          1313  2     sysoutfab : REF $BBLOCK,
;   733          1314  2     sysoutnam : REF $BBLOCK;
;   734          1315  2
;   735          1316  2 LOCAL
;   736          1317  2     sysoutstring : VECTOR [nam$c_maxrss, BYTE],                ! Space for SYS$OUTPUT resultant filename
;   737          1318  2     status;
;   738          1319  2
;   739        P 1320  2 $NAM_INIT (       NAM = .sysoutnam,
;   740        P 1321  2                   ESS = nam$c_maxrss,
;   741        P 1322  2                   ESA = sysoutstring,
;   742        P 1323  2                   RSS = nam$c_maxrss,
;   743          1324  2                   RSA = sysoutstring);
;   744          1325  2
;   745        P 1326  2 $FAB_INIT (       FAB = .sysoutfab,
;   746        P 1327  2                   FNS = .sysoutdesc [dsc$w_length],
;   747        P 1328  2                   FNA = .sysoutdesc [dsc$a_pointer],
;   748        P 1329  2                   DNM = 'SYS$DISK:HELP.LIS',
;   749        P 1330  2                   RAT = CR,
;   750        P 1331  2                   FAC = PUT,
;   751          1332  2                   NAM = .sysoutnam );
;   752          1333  2
;   753        P 1334  2 $RAB_INIT (       RAB = sysoutrab,
;   754          1335  2                   FAB = .sysoutfab );
;   755          1336  2
;   756          1337  3 IF NOT (status = $CREATE (FAB = .sysoutfab))
;   757          1338  3    THEN BEGIN
;   758          1339  3         sysout_name [dsc$w_length] = .sysoutnam [nam$b_esl];
;   759          1340  3         sysout_name [dsc$a_pointer] = .sysoutnam [nam$l_esa];
;   760          1341  3         SIGNAL_STOP ( (shr$_openout OR hlp$c_facility OR sts$k_error),
;   761          1342  3                       1, sysout_name, .status, .lbr$gl_rmsstv );
;   762          1343  3         END;
;   763          1344  2
```

```
;  764          1345  2 sysout_name [dsc$w_length] = .sysoutnam [nam$b_rsl];
;  765          1346  2 sysout_name [dsc$a_pointer] = .sysoutnam [nam$l_rsa];
;  766          1347  2
;  767          1348  3 IF NOT (status = $CONNECT (RAB = sysoutrab))
;  768          1349  2    THEN SIGNAL_STOP ( (shr$_openout OR hlp$c_facility OR sts$k_error),
;  769          1350  2                       1, sysout_name, .status, .sysoutrab [rab$l_stv] );
;  770          1351  2
;  771          1352  2 RETURN true;
;  772          1353  1 END;


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

4C 2E 50 4C 45 48 3A 4B 53 49 44 24 53 59 53  0010B P.ABF:  .ASCII  \SYS$DISK:HELP.LIS\                          ;
                                              53 49  0011A                                                      ;

                                                              $RMS_PTR=         SYSOUTRAB
                                                        .EXTRN  SYS$CREATE, SYS$CONNECT

                                                        .PSECT  $CODE$,NOWRT,2

                                                03FC 00000 OPEN_SYSOUTPUT:
                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9                    ; 1288
                                  59 00000000G   00  9E 00002           MOVAB   LIB$STOP, R9                    ;
                                  58 00000000'   EF  9E 00009           MOVAB   SYSOUT_NAME, R8                 ;
                                  5E      FF00   CE  9E 00010           MOVAB   -256(SP), SP                    ;
                                  57        08   AC  D0 00015           MOVL    SYSOUTNAM, R7                   ; 1324
    0060   8F            00      6E             00  2C 00019           MOVC5   #0, (SP), #0, #96, (R7)         ;
                                                  67     00020          ;
                                  67      6002   8F  B0 00021           MOVW    #24578, (R7)                    ;
                            02    A7             01  8E 00026           MNEGB   #1, 2(R7)                       ;
                            04    A7             6E  9E 0002A           MOVAB   SYSOUTSTRING, 4(R7)             ;
                            0A    A7             01  8E 0002E           MNEGB   #1, 10(R7)                      ;
                            0C    A7             6E  9E 00032           MOVAB   SYSOUTSTRING, 12(R7)            ;
                                  56        04   AC  D0 00036           MOVL    SYSOUTFAB, R6                   ; 1332
0050    8F            00      6E             00  2C 0003A           MOVC5   #0, (SP), #0, #80, (R6)         ;
                                                  66     00041          ;
                                  66      5003   8F  B0 00042           MOVW    #20483, (R6)                    ;
                            16    A6             01  90 00047           MOVB    #1, 22(R6)                      ;
                            1E    A6      0202   8F  B0 0004B           MOVW    #514, 30(R6)                    ;
                            28    A6             57  D0 00051           MOVL    R7, 40(R6)                      ;
                            2C    A6        0C   A8  D0 00055           MOVL    SYSOUTDESC+4, 44(R6)            ;
                            30    A6 00000000'   EF  9E 0005A           MOVAB   P.ABF, 48(R6)                   ;
                            34    A6        08   A8  90 00062           MOVB    SYSOUTDESC, 52(R6)              ;
                            35    A6             11  90 00067           MOVB    #17, 53(R6)                     ;
0044    8F            00      6E             00  2C 0006B           MOVC5   #0, (SP), #0, #68, $RMS_PTR     ; 1335
                                        BC        A8  00072              ;
                            BC    A8      4401   8F  B0 00074           MOVW    #17409, $RMS_PTR                ;
                            F8    A8             56  D0 0007A           MOVL    R6, $RMS_PTR+60                 ;
                                                 56  DD 0007E           PUSHL   R6                              ; 1337
                      00000000G   00             01  FB 00080           CALLS   #1, SYS$CREATE                  ;
                                  52             50  D0 00087           MOVL    R0, STATUS                      ;
                                  1E             52  E8 0008A           BLBS    STATUS, 1$                      ;
                                  68        0B   A7  9B 0008D           MOVZBW  11(R7), SYSOUT_NAME             ; 1339
                            04    A8        0C   A7  D0 00091           MOVL    12(R7), SYSOUT_NAME+4           ; 1340
                                  00000000G      00  DD 00096           PUSHL   LBR$GL_RMSSTV                   ; 1342
```

```
                                 52  DD 0009C          PUSHL   STATUS                                                    ;
                                 58  DD 0009E          PUSHL   R8                                                      ; 1341
                                 01  DD 000A0          PUSHL   #1                                                      ;
                      007610A2   8F  DD 000A2          PUSHL   #7737506                                                ;
                             69  05  FB 000A8          CALLS   #5, LIB$STOP                                            ;
                             68      03  A7  9B 000AB 1$:     MOVZBW  3(R7), SYSOUT_NAME                              ; 1345
                          04 A8      04  A7  D0 000AF          MOVL    4(R7), SYSOUT_NAME+4                            ; 1346
                                     BC  A8  9F 000B4          PUSHAB  SYSOUTRAB                                       ; 1348
                00000000G    00      01  FB 000B7          CALLS   #1, SYS$CONNECT                                 ;
                             52      50  D0 000BE          MOVL    R0, STATUS                                      ;
                             12      52  E8 000C1          BLBS    STATUS, 2$                                      ;
                                 C8  A8  DD 000C4          PUSHL   SYSOUTRAB+12                                    ; 1350
                                 52  DD 000C7          PUSHL   STATUS                                          ;
                                 58  DD 000C9          PUSHL   R8                                              ; 1349
                                 01  DD 000CB          PUSHL   #1                                              ;
                      007610A2   8F  DD 000CD          PUSHL   #7737506                                        ;
                             69  05  FB 000D3          CALLS   #5, LIB$STOP                                    ;
                             50      01  D0 000D6 2$:     MOVL    #1, R0                                          ; 1352
                                     04  000D9          RET                                                     ; 1353

; Routine Size:  218 bytes,    Routine Base:  $CODE$ + 0511
```

```
;   774          1354  1 ROUTINE clean_up (sysoutfab) =
;   775          1355  2 BEGIN
;   776          1356  2
;   777          1357  2 !++
;   778          1358  2 ! FUNCTIONAL DESCRIPTION:
;   779          1359  2 !
;   780          1360  2 !       Deassign SYS$INPUT if assigned and disconnect and close output file.
;   781          1361  2 !
;   782          1362  2 ! INPUTS:
;   783          1363  2 !
;   784          1364  2 !       sysoutfab =     address of FAB for SYS$OUTPUT
;   785          1365  2 !
;   786          1366  2 ! OUTPUTS:
;   787          1367  2 !
;   788          1368  2 !       None.
;   789          1369  2 !
;   790          1370  2 ! ROUTINE VALUE:
;   791          1371  2 !
;   792          1372  2 !       Always true.
;   793          1373  2 !
;   794          1374  2 !--
;   795          1375  2
;   796          1376  2 MAP
;   797          1377  2     sysoutfab : REF $BBLOCK;
;   798          1378  2
;   799          1379  2 LOCAL
;   800          1380  2     status;
;   801          1381  2
;   802          1382  3 IF (.help_flags [hlp$v_prompt] OR .help_flags [hlp$v_page])
;   803          1383  2    AND NOT .help_flags [hlp$v_notterm]
;   804          1384  3    THEN IF NOT (status = $DASSGN (CHAN = .sysinchan))
;   805          1385  2                THEN SIGNAL (.status);
;   806          1386  2
;   807          1387  3 IF NOT (status = $DISCONNECT (RAB = sysoutrab))
;   808          1388  2    THEN SIGNAL ( (shr$_closeout OR hlp$c_facility OR sts$k_warning),
;   809          1389  2         1, sysout_name, .status, .sysoutrab [rab$l_stv]);
;   810          1390  2
;   811          1391  3 IF NOT (status = $CLOSE (FAB = .sysoutfab))
;   812          1392  2    THEN SIGNAL ( (shr$_closeout OR hlp$c_facility OR sts$k_warning),
;   813          1393  2         1, sysout_name, .status, .sysoutrab [rab$l_stv]);
;   814          1394  2
;   815          1395  2 RETURN true;
;   816          1396  1 END;


                                                        .EXTRN  SYS$DISCONNECT, SYS$CLOSE

                                        001C 00000 CLEAN_UP:
                                                        .WORD   Save R2,R3,R4                   ; 1354
                        54 00000000G    00   9E 00002        MOVAB   LIB$SIGNAL, R4                  ;
                        53 00000000'    EF   9E 00009        MOVAB   HELP_FLAGS, R3                  ;
                        04              63   E8 00010        BLBS    HELP_FLAGS, 1$                  ; 1382
                        1B        01    A3   E9 00013        BLBC    HELP_FLAGS+1, 2$                ;
                16        01    A3            06   E0 00017 1$:     BBS     #6, HELP_FLAGS+1, 2$    ; 1383
                                    FD98      C3   DD 0001C        PUSHL   SYSINCHAN               ; 1384
                   00000000G    00            01   FB 00020        CALLS   #1, SYS$DASSGN          ;
```

```
                                52              50   D0 00027          MOVL    R0, STATUS                              ;
                                05              52   E8 0002A          BLBS    STATUS, 2$                              ;
                                                52   DD 0002D          PUSHL   STATUS                                  ; 1385
                                64              01   FB 0002F          CALLS   #1, LIB$SIGNAL                          ;
                                        FD9C    C3   9F 00032 2$:      PUSHAB  SYSOUTRAB                               ; 1387
                00000000G       00              01   FB 00036          CALLS   #1, SYS$DISCONNECT                      ;
                                52              50   D0 0003D          MOVL    R0, STATUS                              ;
                                15              52   E8 00040          BLBS    STATUS, 3$                              ;
                                        FDA8    C3   DD 00043          PUSHL   SYSOUTRAB+12                            ; 1389
                                                52   DD 00047          PUSHL   STATUS                                  ;
                                        FDE0    C3   9F 00049          PUSHAB  SYSOUT_NAME                             ; 1388
                                                01   DD 0004D          PUSHL   #1                                      ;
                                00761058        8F   DD 0004F          PUSHL   #7737432                                ;
                                64              05   FB 00055          CALLS   #5, LIB$SIGNAL                          ;
                                        04      AC   DD 00058 3$:      PUSHL   SYSOUTFAB                               ; 1391
                00000000G       00              01   FB 0005B          CALLS   #1, SYS$CLOSE                           ;
                                52              50   D0 00062          MOVL    R0, STATUS                              ;
                                15              52   E8 00065          BLBS    STATUS, 4$                              ;
                                        FDA8    C3   DD 00068          PUSHL   SYSOUTRAB+12                            ; 1393
                                                52   DD 0006C          PUSHL   STATUS                                  ;
                                        FDE0    C3   9F 0006E          PUSHAB  SYSOUT_NAME                             ; 1392
                                                01   DD 00072          PUSHL   #1                                      ;
                                00761058        8F   DD 00074          PUSHL   #7737432                                ;
                                64              05   FB 0007A          CALLS   #5, LIB$SIGNAL                          ;
                                50              01   D0 0007D 4$:      MOVL    #1, R0                                  ; 1395
                                                     04 00080          RET                                             ; 1396

; Routine Size:  129 bytes,    Routine Base:  $CODE$ + 05EB
```

```
;  818     1397  1 %SBTTL 'Routine make_upper_case';
;  819     1398  1 ROUTINE make_upper_case (idesc, oname) =
;  820     1399  2 BEGIN
;  821     1400  2
;  822     1401  2 !++
;  823     1402  2 ! FUNCTIONAL DESCRIPTION:
;  824     1403  2 !
;  825     1404  2 !       Upper case the name described by string descriptor idesc and
;  826     1405  2 !       put the name at location oname.
;  827     1406  2 !
;  828     1407  2 ! INPUTS:
;  829     1408  2 !
;  830     1409  2 !       idesc =         address of string descriptor for input text string
;  831     1410  2 !
;  832     1411  2 !       oname =         address of buffer to contain uppercase output string
;  833     1412  2 !
;  834     1413  2 ! OUTPUTS:
;  835     1414  2 !
;  836     1415  2 !       oname : as described above
;  837     1416  2 !
;  838     1417  2 ! ROUTINE VALUE:
;  839     1418  2 !
;  840     1419  2 !       Always true.
;  841     1420  2 !
;  842     1421  2 !--
;  843     1422  2
;  844     1423  2 MAP
;  845     1424  2     idesc : REF $BBLOCK,
;  846     1425  2     oname : REF VECTOR[,BYTE];
;  847     1426  2
;  848     1427  2 BIND
;  849     1428  2     namlen = idesc[dsc$w_length] : WORD,
;  850     1429  2     iname = idesc[dsc$a_pointer] : REF VECTOR[,BYTE];
;  851     1430  2
;  852     1431  2 IF .namlen GTRU 0                                   ! If non-empty string
;  853     1432  2    THEN INCRU i FROM 0 TO .namlen-1                 ! Then for each character
;  854     1433  2
;  855     1434  2 DO IF .iname[.i] GEQU %ASCII'a'                     ! Convert character to uppercase and copy
;  856     1435  2       AND .iname[.i] LEQU %ASCII'z'
;  857     1436  3       THEN oname[.i] = .iname[.i] - (%ASCII'a' - %ASCII'A')
;  858     1437  2       ELSE IF .iname[.i] EQL 9
;  859     1438  2               THEN oname[.i] = 32
;  860     1439  2               ELSE oname[.i] = .iname[.i];
;  861     1440  2
;  862     1441  2 RETURN true
;  863     1442  2
;  864     1443  1 END;                                               !Of make_upper_case


                        001C 00000 MAKE_UPPER_CASE:
                                   .WORD   Save R2,R3,R4                ; 1398
 53      04   AC                   ADDL3   #4, IDESC, R3                ; 1429
                    04  BC  B5 00007  TSTW    @IDESC                   ; 1431
                        3A  13 0000A  BEQL    6$                       ;
```

```
                                54        04    BC   3C 0000C          MOVZWL  @IDESC, R4                                  ; 1432
                                          54    D7   00010             DECL    R4
                                          52    D4   00012             CLRL    I                                         ; 1436
                                          2B    11   00014             BRB     5$
                     51                   52    08   AC   C1 00016 1$:  ADDL3   ONAME, I, R1
                                          50    00 B342   9A 0001B      MOVZBL  @0(R3)[I], R0                             ; 1434
                              61          8F          50   91 00020     CMPB    R0, #97
                                          0C          1F   00024       BLSSU   2$
                              7A          8F          50   91 00026     CMPB    R0, #122                                  ; 1435
                                          06          1A   0002A       BGTRU   2$
                     61                   50          20   83 0002C     SUBB3   #32, R0, (R1)                             ; 1436
                                          0D          11   00030       BRB     4$
                                09        50   91 00032 2$:  CMPB    R0, #9                                    ; 1437
                                          05    12   00035             BNEQ    3$
                                61        20   90 00037               MOVB    #32, (R1)                                 ; 1438
                                          03    11   0003A             BRB     4$
                                61        50   90 0003C 3$:  MOVB    R0, (R1)                                  ; 1439
                                          52    D6   0003F 4$:  INCL    I                                         ; 1434
                                54        52   D1 00041 5$:  CMPL    I, R4
                                          D0    1B   00044             BLEQU   1$
                                50        01   D0 00046 6$:  MOVL    #1, R0                                    ; 1441
                                               04 00049               RET                                       ; 1443

; Routine Size:  74 bytes,    Routine Base:  $CODE$ + 066C

;  865          1444  1
;  866          1445  1 END
;  867          1446  0 ELUDOM                                                !Of module


                                                                    .EXTRN  LIB$SIGNAL, LIB$STOP

;                                PSECT SUMMARY

;      Name                      Bytes                          Attributes

;  $PLIT$                         284  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                          620  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                        1718  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       135         1         581        00:01.1


; Information:  3
```

```
; Warnings:       0
; Errors:         0


;                                       COMMAND QUALIFIERS

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

; Size:           1718 code + 904 data bytes
; Run Time:         00:36.2
; Elapsed Time:     01:25.3
; Lines/CPU Min:     2398
; Lexemes/CPU-Min: 31759
; Memory Used:  221 pages
; Compilation Complete
```

HELP

FORWRITDF
LIS

VMSHELP
MAP

HLD
MDL

FORWRITDU
LIS

FORVECTOR
LIS

FORWRITIL
LIS

FORWRITSF
LIS

HLD

HLDMACROS
MAR

FORVM
LIS

FORWRITSN
LIS

HLD
MAP

FORWRITSU
LIS

RSXDEFS
MAR

FORUNLOCK
LIS

FORWRITDO
LIS

HELP
LIS

FORWRITIF
LIS

FORWRITIO
LIS

HLDCSTA
LIS

FORUDFWU
LIS

FORUNDERF
LIS

FORWRITSL
LIS

FORWRITSO
LIS