```
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                 IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLL                    III        BBB         BBB  RRR      RRR           TTT        LLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL     IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

```
LL            IIIIII    BBBBBBBB      SSSSSSSS    IIIIII      GGGGGGGG    SSSSSSSS  TTTTTTTTTT    OOOOOO
LL            IIIIII    BBBBBBBB      SSSSSSSS    IIIIII      GGGGGGGG    SSSSSSSS  TTTTTTTTTT    OOOOOO
LL              II      BB      BB  SS              II      GG            SS            TT        OO      OO
LL              II      BB      BB  SS              II      GG            SS            TT        OO      OO
LL              II      BB      BB  SS              II      GG            SS            TT        OO      OO
LL              II      BB      BB  SS              II      GG            SS            TT        OO      OO
LL              II      BBBBBBBB      SSSSSS        II      GG              SSSSSS      TT        OO      OO
LL              II      BBBBBBBB      SSSSSS        II      GG              SSSSSS      TT        OO      OO
LL              II      BB      BB            SS    II      GG  GGGGGG            SS  TT        OO      OO
LL              II      BB      BB            SS    II      GG  GGGGGG            SS  TT        OO      OO
LL              II      BB      BB            SS    II      GG      GG            SS  TT        OO      OO
LL              II      BB      BB            SS    II      GG      GG            SS  TT        OO      OO
LLLLLLLLLL    IIIIII    BBBBBBBB    SSSSSSSS    IIIIII      GGGGGG    SSSSSSSS      TT          OOOOOO
LLLLLLLLLL    IIIIII    BBBBBBBB    SSSSSSSS    IIIIII      GGGGGG    SSSSSSSS      TT          OOOOOO


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

Table of contents

```
0000     1          .TITLE  LIBSSIGSTOP     Convert a signal to a stop
0000     2          .IDENT  /1-001/          ; File: LIBSIGSTO.MAR  EDIT: FM1001
0000     3
0000     4  ;
0000     5  ;*****************************************************************************
0000     6  ;*                                                                           *
0000     7  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000     8  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                   *
0000     9  ;*  ALL RIGHTS RESERVED.                                                     *
0000    10  ;*                                                                           *
0000    11  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    12  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    13  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    14  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    15  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    16  ;*  TRANSFERRED.                                                             *
0000    17  ;*                                                                           *
0000    18  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    19  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    20  ;*  CORPORATION.                                                             *
0000    21  ;*                                                                           *
0000    22  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    23  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    24  ;*                                                                           *
0000    25  ;*                                                                           *
0000    26  ;*****************************************************************************
0000    27  ;
0000    28  ;++
0000    29  ; FACILITY: Common Run-time library.
0000    30  ;
0000    31  ; AUTHOR:
0000    32  ;       FAROKH MORSHED 22-APR-83 VMS V3B
0000    33  ;
0000    34  ; MODIFIED BY:
0000    35  ;
0000    36  ;--
0000    37
0000    38  ;
0000    39  ; PSECT DEFINITION
0000    40  ;
00000000    41          .PSECT _LIB$CODE PIC, SHR, LONG, NOWRT
0000    42
0000    43  ;
0000    44  ; MACRO LIBRARY CALLS
0000    45  ;
0000    46          $CHFDEF                         ; Condition Handling arglst offsets
0000    47          $STSDEF                         ; Status code fields
0000    48
0000    49  ;
0000    50  ; EXTERNAL REFERENCES
0000    51  ;
0000    52          .DSABL          GBL
0000    53          .EXTERNAL       LIB$_INVARG     ; We return this error condition
0000    54                                          ; when SIGARGS condition code is
0000    55                                          ; SS$_UNWIND.
0000    56          .EXTERNAL       SS$_UNWIND      ; Unwind condition code
0000    57          .EXTERNAL       SS$_NORMAL      ; Successful completion status
```

```
                     59          .SBTTL  Change a signal to a stop
                     60
                     61 ;++
                     62 ; FUNCTIONAL DESCRIPTION:
                     63 ;       This routine sets the stop bit associated with the exception whose
                     64 ;       mechanism argument is passed.  The mechanism argument address is used
                     65 ;       in a "dead reckon" fashion to find the flag longword.  The signal
                     66 ;       argument address is used to set the severity of the condition to
                     67 ;       severe.  This routine is used to convert a particular signal to
                     68 ;       appear as if it was signalled by a LIB$STOP.  If the condition value
                     69 ;       in SIGARGS is SS$_UNWIND, then this routine returns an error condition.
                     70 ;       The SS$_UNWIND condition code is compared with the SIGARGS condition
                     71 ;       code in a fashion that is only "correct" when a non-facility specific
                     72 ;       condition code (which SS$_UNWIND indeed is one) is being compared
                     73 ;       against any other condition code (this is to avoid the overhead of
                     74 ;       calling LIB$MATCH_COND).
                     75 ;
                     76 ; CALLING SEQUENCE:
                     77 ;       LIB$SIG_TO_STOP (SIGARGS.mr.r, MCHARGS.rr.r)
                     78 ;
                     79 ; FORMAL PARAMETERS:
                     80 ;       SIGARGS         The signal's signal argument
                     81 ;       MCHARGS         The signal's mechanism argument
                     82 ;
                     83 ; IMPLICIT INPUTS:
                     84 ;       NONE
                     85 ;
                     86 ; IMPLICIT OUTPUTS:
                     87 ;       NONE
                     88 ;
                     89 ; COMPLETION CODES:
                     90 ;       SS$_NORMAL              ; Success
                     91 ;       LIB$_INVARG             ; The condition code is SS$_UNWIND
                     92 ;
                     93 ; SIDE EFFECTS:
                     94 ;
                     95 ;--
                     96
          00000014   97          MCHARGS_TO_FLAGS_OFFSET = 20            ; Dead reckon offset to flag
          00000001   98          V_STOP_BIT      = 1                     ; Position of stop bit
          00000002   99          M_STOP_BIT      = 1@V_STOP_BIT          ; Mask for stop bit
          00000000  100          V_SIGNAL_BIT    = 0                     ; Pos of signal bit
          00000001  101          M_SIGNAL_BIT    = 1@V_SIGNAL_BIT        ; Mask for signal bit
                    102
                    103 UNWIND_MSG:
          00000000' 104          .LONG SS$_UNWIND@-STS$S_SEVERITY; The message portion of SS$_UNWIND
                    105
              0000  106          .ENTRY LIB$SIG_TO_STOP, ^M<>
                    107
    50  04 AC  D0   108          MOVL    CHF$L_SIGARGLST(AP), R0         ; R0 <- SIGARGS
                    109
        0D  03 ED   110          CMPZV   #STS$V_MSG_NO, #STS$S_MSG_NO,-
EF AF   04 A0       111                  CHF$L_SIG_NAME(R0), UNWIND_MSG  ; SS$_UNWIND?
            1B  13  112          BEQL    10$                             ; Yes.
                    113
    00  04  F0   114          INSV    #STS$K_SEVERE, #STS$V_SEVERITY,-
04 A0   03       115                  #STS$S_SEVERITY,CHF$L_SIG_NAME(R0); Set severity to severe
```

```
                              0019
        50   08 AC      D0    0019             MOVL    CHF$L_MCHARGLST(AP), R0              ; R0 <- MCHARGS
        14 A0    02     88    001D             BISB2   #M_STOP_BIT, MCHARGS_TO_FLAGS_OFFSET(R0)
                              0021                                                          ; Set the stop bit
        14 A0    01     8A    0021             BICB2   #M_SIGNAL_BIT, MCHARGS_TO_FLAGS_OFFSET(R0)
                              0025                                                          ; Clear signal bit
   50   00000000'8F     D0    0025             MOVL    #SS$_NORMAL, R0                      ; Success
                 07     11    002C             BRB     20$                                  ;  and return
                              002E      10$:
   50   00000000'8F     D0    002E             MOVL    #LIB$_INVARG, R0                     ; Condition code is SS$_UNWIND
                              0035      20$:
                        04    0035             RET                                          ;
                              0036
                              0036             .END
```

```
CHF$L_MCHARGLST                 = 00000008
CHF$L_SIGARGLST                 = 00000004
CHF$L_SIG_NAME                  = 00000004
LIB$SIG_TO_STOP                   00000004 RG     01
LIB$_INVARG                       ********   X    01
MCHARGS_TO_FLAGS_OFFSET         = 00000014
M_SIGNAL_BIT                    = 00000001
M_STOP_BIT                      = 00000002
SS$_NORMAL                        ********   X    01
SS$_UNWIND                        ********   X    01
STS$K_SEVERE                    = 00000004
STS$S_MSG_NO                    = 0000000D
STS$S_SEVERITY                  = 00000003
STS$V_MSG_NO                    = 00000003
STS$V_SEVERITY                  = 00000000
UNWIND_MSG                        00000000 R      01
V_SIGNAL_BIT                    = 00000000
V_STOP_BIT                      = 00000001
```

Psect synopsis

PSECT name	Allocation	PSECT No.	Attributes
. ABS .	00000000 (0.)	00 (0.)	NOPIC USR CON ABS LCL NOSHR NOEXE NORD NOWRT NOVEC BYTE
LIB$CODE	00000036 (54.)	01 (1.)	PIC USR CON REL LCL SHR EXE RD NOWRT NOVEC LONG
ABS	00000000 (0.)	02 (2.)	NOPIC USR CON ABS LCL NOSHR EXE RD WRT NOVEC BYTE

Performance indicators

Phase	Page faults	CPU Time	Elapsed Time
Initialization	29	00:00:00.03	00:00:01.46
Command processing	112	00:00:00.34	00:00:04.08
Pass 1	122	00:00:00.77	00:00:06.69
Symbol table sort	0	00:00:00.02	00:00:00.02
Pass 2	39	00:00:00.22	00:00:01.47
Symbol table output	3	00:00:00.01	00:00:00.01
Psect synopsis output	3	00:00:00.02	00:00:00.02
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	310	00:00:01.41	00:00:13.75

```
The working set limit was 900 pages.
4230 bytes (9 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 55 non-local and 2 local symbols.
129 source lines were read in Pass 1, producing 13 object records in Pass 2.
9 pages of virtual memory were used to define 8 macros.
```

```
                                    +----------------------------+
                                    ! Macro library statistics !
                                    +----------------------------+

Macro library name                                  Macros defined
------------------                                  --------------
_$255$DUA28:[SYSLIB]STARLET.MLB;2                            5

108 GETS were required to define 5 macros.

There were no errors, warnings or information messages.

MACRO/ENABLE=SUPPRESSION/DISABLE=(GLOBAL,TRACEBACK)/LIS=LIS$:LIBSIGSTO/OBJ=OBJ$:LIBSIGSTO MSRC$:LIBSIGSTO/UPDATE=(ENH$:LIBSIGSTO)
```

0209 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

LIBPOLYG
LIS

LIBREMQHI
LIS

LIBRENAME
LIS

LIBSCANC
LIS

LIBSIGSTO
LIS

LIBROOBJ
LIS

LIBRUNPRO
LIS

LIBSIGNAL
LIS

LIBPUTOUT
LIS

LIBREMQTI
LIS

LIBSIGRET
LIS

LIBSIMTRA
LIS

LIBSCOPY
LIS

LIBPOLYH
LIS

LIBREVERT
LIS