```
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
```

FILEID**LIBEF

```
LL            IIIIII    BBBBBBBB    EEEEEEEEEE  FFFFFFFFFF
LL            IIIIII    BBBBBBBB    EEEEEEEEEE  FFFFFFFFFF
LL              II      BB      BB  EE          FF
LL              II      BB      BB  EE          FF
LL              II      BB      BB  EE          FF
LL              II      BB      BB  EE          FF
LL              II      BBBBBBBB    EEEEEEEE    FFFFFFFF
LL              II      BBBBBBBB    EEEEEEEE    FFFFFFFF
LL              II      BB      BB  EE          FF
LL              II      BB      BB  EE          FF
LL              II      BB      BB  EE          FF               ....
LL              II      BB      BB  EE          FF               ....
LLLLLLLLLL    IIIIII    BBBBBBBB    EEEEEEEEEE  FF               ....
LLLLLLLLLL    IIIIII    BBBBBBBB    EEEEEEEEEE  FF               ....

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


```
(2)     57    DECLARATIONS
(3)    115    LIB$GET_EF - Allocate one local event flag
(4)    210    LIB$FREE_EF - Deallocate one local event flag
(5)    320    LIB$RESERVE_EF - Reserve a local event flag
```

```
0000        .TITLE  LIB$EF - Resource allocator for local event flags
0000        .IDENT  /1-006/                  ; File: LIBEF.MAR  Edit: MDL1006
0000
0000  ;
0000  ;****************************************************************************
0000  ;*                                                                          *
0000  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
0000  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
0000  ;*  ALL RIGHTS RESERVED.                                                    *
0000  ;*                                                                          *
0000  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
0000  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
0000  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
0000  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
0000  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
0000  ;*  TRANSFERRED.                                                            *
0000  ;*                                                                          *
0000  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
0000  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
0000  ;*  CORPORATION.                                                            *
0000  ;*                                                                          *
0000  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
0000  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
0000  ;*                                                                          *
0000  ;*                                                                          *
0000  ;****************************************************************************
0000  ;
0000
0000  ;++
0000  ; FACILITY: General Utility Library
0000  ;
0000  ; ABSTRACT:
0000  ;
0000  ;       Three routines for allocating and deallocating local event
0000  ;       flag numbers.  Using these routines allows use of local
0000  ;       event flags by multiple procedures without conflicts.
0000  ;
0000  ; ENVIRONMENT: User Mode, AST Reentrant
0000  ;
0000  ;--
0000  ; AUTHOR: Steven B. Lionel, CREATION DATE: 08-DEC-78
0000  ;
0000  ; MODIFIED BY:
0000  ;
0000  ; SBL, : VERSION 1
0000  ; 1-001 - Original
0000  ; 1-002 - Pre-reserve event flags 1-23.  SBL 18-Dec-78
0000  ; 1-003 - Put _ before PSECT names and make error codes be
0000  ;           global.  JBS for SBL 23-JAN-1979
0000  ; 1-004 - Correct a typo in edit 003.  JBS 24-JAN-1979
0000  ; 1-005 - Add a first time flag to allow initialization of non-zero data at
0000  ;         run-time, thus allowing the linker to perform demand-zero compression
0000  ;         and helping performance.  MDL 6-Jul-1984
0000  ; 1-006 - Change interpretation of 1/2 the bits in EF_POOL, eliminating the need for
0000  ;         a first time flag and restoring AST re-entrancy.  MDL 7-Aug-1984
```

```
                      0000          .SBTTL  DECLARATIONS
                      0000  ;
                      0000  ; INCLUDE FILES:
                      0000  ;
                      0000
                      0000  ;
                      0000  ; EXTERNAL DECLARATIONS:
                      0000  ;
                      0000          .DSABL  GBL                       ; Prevent undeclared
                      0000                                            ; symbols from being
                      0000                                            ; automatically global.
                      0000          .EXTRN  LIB$_INSEF                ; Error codes
                      0000          .EXTRN  LIB$_EF_ALRRES
                      0000          .EXTRN  LIB$_EF_ALRFRE
                      0000          .EXTRN  LIB$_EF_RESSYS
                      0000
                      0000  ;
                      0000  ; MACROS:
                      0000  ;
                      0000
                      0000  ;
                      0000  ; EQUATED SYMBOLS:
                      0000  ;
          00000004    0000          ef_number = 4         ; event flag number parameter
                      0000  ;
                      0000  ; OWN STORAGE:
                      0000  ;
          00000000    0000          .PSECT  _LIB$DATA RD, WRT, NOEXE, NOSHR, PIC, LONG, -
                      0000                          CON, LCL, REL, USR
                      0000
                      0000  EF_POOL:                      ; Pool of available event flags
 00000000 00000000    0000          .QUAD   0
                      0008
                      0008          ; P L E A S E   N O T E   ! ! !
                      0008          ;
                      0008          ; EF_POOL is structured as follows:
                      0008          ; Low order bit (bit 0) represents event flag number 63.
                      0008          ; High order bit (bit 63) represents event flag number 0.
                      0008          ;
                      0008          ; Bits 0-31 of EF_POOL (event flags 32-63) are interpreted as folows:
                      0008          ; 0 => available
                      0008          ; 1 => reserved
                      0008          ; Bits 32-63 of EF_POOL (event flags 0-31) are interpreted as follows:
                      0008          ; 0 => reserved
                      0008          ; 1 => available
                      0008          ;
                      0008          ; Thus, the initial setting of 0 allows for:
                      0008          ; Event flags 0 and  24-31 are reserved to the system.
                      0008          ; Event flags 1-23 are pre-reserved.
                      0008          ; Event flags 32-63 are available for use.
                      0008
                      0008  ;
                      0008  ; PSECT DECLARATIONS:
                      0008  ;
                      0008          .PSECT _LIB$CODE PIC, USR, CON, REL, LCL, SHR, -
          00000000    0008                          EXE, RD, NOWRT, LONG
                      0000
```

```
                                    0000          .SBTTL  LIB$GET_EF - Allocate one local event flag
                                    0000    ;++
                                    0000    ; FUNCTIONAL DESCRIPTION:
                                    0000    ;
                                    0000    ;       LIB$GET_EF allocates one local event flag from a process-wide
                                    0000    ;       pool.  If a flag is available for use, its number is returned
                                    0000    ;       to the caller.  If no flags are available, an error is returned
                                    0000    ;       as the function value.
                                    0000    ;
                                    0000    ; CALLING SEQUENCE:
                                    0000    ;
                                    0000    ;       status.wlc.v = LIB$GET_EF (ef_number.wl.r)
                                    0000    ;
                                    0000    ; INPUT PARAMETERS:
                                    0000    ;
                                    0000    ;       NONE
                                    0000    ;
                                    0000    ; IMPLICIT INPUTS:
                                    0000    ;
                                    0000    ;       EF_POOL, a table of available event flag numbers in OWN
                                    0000    ;       storage.
                                    0000    ;
                                    0000    ; OUTPUT PARAMETERS:
                                    0000    ;
                                    0000    ;       ef_number       - The number of the local event flag allocated
                                    0000    ;                         or -1 if none were available.
                                    0000    ;
                                    0000    ; IMPLICIT OUTPUTS:
                                    0000    ;
                                    0000    ;       If successful, an entry is made into EF_POOL indicating that
                                    0000    ;       a local event flag has been reserved.
                                    0000    ;
                                    0000    ; FUNCTION VALUE:
                                    0000    ; COMPLETION CODES:
                                    0000    ;
                                    0000    ;       SS$_NORMAL      - Routine successfully completed.
                                    0000    ;
                                    0000    ;       LIB$_INSEF      - Insufficient event flags.  There were no
                                    0000    ;                         more event flags available for allocation.
                                    0000    ;                         If this error is returned, ef_number is
                                    0000    ;                         also set to -1 in case the caller does not
                                    0000    ;                         check for failure.
                                    0000    ;
                                    0000    ; SIDE EFFECTS:
                                    0000    ;
                                    0000    ;       NONE
                                    0000    ;
                                    0000    ;--
                                    0000
                              4000  0000          .ENTRY LIB$GET_EF, ^M<IV>       ; Save nothing
                                    0002
                                    0002    ;+
                                    0002    ;       Scan EF_POOL for first available event flag number
                                    0002    ;-
                                    0002
                                    0002    SCAN:
  50   00000000'EF   20   00   EB   0002          FFC     #0, #32, EF_POOL, R0
```

```
                             15   12 000B           BNEQ    FOUND                   ; One was found
   50  00000000'EF    20     20   EA 000D           FFS     #32, #32, EF_POOL, R0   ; Try next 32 positions
                             1B   13 0016           BEQL    ALL_OUT                 ; None available
                                     0018
                                     0018   ;+
                                     0018   ;       Now recheck and set the bit under interlock, in case someone
                                     0018   ;       has set it at AST level in the meantime.
                                     0018   ;-
                                     0018
                                     0018   FOUND_LOW:
        E2 00000000'EF       50   E5 0018           BBCC    R0, EF_POOL, SCAN       ; Repeat scan if already set
                             08   11 0020           BRB     LEAVE_GET
                                     0022
                                     0022   FOUND:
        D8 00000000'EF       50   E2 0022           BBSS    R0, EF_POOL, SCAN       ; Repeat scan if already set
                                     002A
                                     002A   ;+
                                     002A   ;       Return success with event flag number in ef_number.
                                     002A   ;-
                                     002A
                                     002A   LEAVE_GET:
              04 BC   3F     50   C3 002A           SUBL3   R0, #63, @ef_number(AP) ; Subtract from 63 because
                                     002F                                           ; lo order table bit is
                                     002F                                           ; event flag 63.
                                     002F
                      50     01   D0 002F           MOVL    #1, R0                  ; SS$_NORMAL
                                  04 0032           RET                             ; Exit
                                     0033
                                     0033   ;+
                                     0033   ;       Return error since no event flags available
                                     0033   ;-
                                     0033
                                     0033   ALL_OUT:
                 04 BC       01   CE 0033           MNEGL   #1, @ef_number(AP)      ; Set ef-number to -1
        50    00000000'8F         D0 0037           MOVL    #LIB$_INSEF, R0         ; Insufficient event flags
                                  04 003E           RET                             ; Exit
                                     003F
```

```
                        003F          .SBTTL  LIB$FREE_EF - Deallocate one local event flag
                        003F   ;++
                        003F   ; FUNCTIONAL DESCRIPTION:
                        003F   ;
                        003F   ;       LIB$FREE_EF is the complement of LIB$GET_EF.  When a routine
                        003F   ;       called LIB$GET_EF to allocate a local event flag, and no
                        003F   ;       longer needs it, LIB$FREE_EF should be called to free the
                        003F   ;       event flag for use by other routines.
                        003F   ;
                        003F   ; CALLING SEQUENCE:
                        003F   ;
                        003F   ;       status.wlc.v = LIB$FREE_EF (ef_number.rl.r)
                        003F   ;
                        003F   ; INPUT PARAMETERS:
                        003F   ;
                        003F   ;       ef_number       - The number of the event flag to be
                        003F   ;                         deallocated.  This is the value returned
                        003F   ;                         to the user by LIB$GET_EF.
                        003F   ;
                        003F   ; IMPLICIT INPUTS:
                        003F   ;
                        003F   ;       EF_POOL, a table of available event flag numbers in OWN
                        003F   ;       storage.
                        003F   ;
                        003F   ; OUTPUT PARAMETERS:
                        003F   ;
                        003F   ;       NONE
                        003F   ;
                        003F   ; IMPLICIT OUTPUTS:
                        003F   ;
                        003F   ;       An entry is made in EF_POOL indicating that the event flag
                        003F   ;       is free for use.
                        003F   ;
                        003F   ; FUNCTION VALUE:
                        003F   ; COMPLETION CODES:
                        003F   ;
                        003F   ;       SS$_NORMAL      - Routine successfully completed.
                        003F   ;
                        003F   ;       LIB$_EF_ALRFRE  - Event flag already free.
                        003F   ;
                        003F   ;       LIB$_EF_RESSYS  - Event flag reserved to system.  This
                        003F   ;                         occurs if ef_number is outside the ranges
                        003F   ;                         of 1-23 and 32-63.
                        003F   ;
                        003F   ; SIDE EFFECTS:
                        003F   ;
                        003F   ;       NONE
                        003F   ;
                        003F   ;--
                        003F
                  4000  003F          .ENTRY LIB$FREE_EF, ^M<IV>      ; Save nothing
                        0041
                        0041   ;+
                        0041   ;       Check to see if ef_number is in the proper range.
                        0041   ;-
                        0041
     3F   04 BC   D1    0041          CMPL    @ef_number(AP), #63     ; Bigger than 63?
```

```
               31   14  0045          BGTR    RES_SYS_1                   ; Yes, error
            04 BC   D5  0047          TSTL    @ef_number(AP)              ; Less than 1?
               2C   15  004A          BLEQ    RES_SYS_1                   ; Yes, error
       17   04 BC   91  004C          CMPB    @ef_number(AP), #23         ; 1-23?
               15   15  0050          BLEQ    OK_LOW                      ; Yes, ok
       20   04 BC   91  0052          CMPB    @ef_number(AP), #32         ; 32-63?
               20   19  0056          BLSS    RES_SYS_1                   ; No, error
                        0058
                        0058     ;+
                        0058     ;        ef_number is in range 32-63.  Now, unset the bit.
                        0058     ;-
                        0058
                        0058     OK_1:
   50   3F   04 BC  C3  0058          SUBL3   @ef_number(AP), #63, R0 ; Convert to bit offset
1B 00000000'EF   50  E5  005D          BBCC    R0, EF_POOL, ALR_FRE        ; Clear but error if
                        0065                                              ; already clear.
               0D   11  0065          BRB     LEAVE_FRE
                        0067
                        0067     ;+
                        0067     ;        ef_number is in range 1-23.  Now, set the bit.
                        0067     ;-
                        0067
                        0067     OK_LOW:
   50   3F   04 BC  C3  0067          SUBL3   @ef_number(AP), #63, R0 ; Convert to bit offset
0C 00000000'EF   50  E2  006C          BBSS    R0, EF_POOL, ALR_FRE        ; SET but error if
                        0074                                              ; already SET.
                        0074
                        0074     ;+
                        0074     ;        Return success
                        0074     ;-
                        0074
                        0074     LEAVE_FRE:
            50   01 D0  0074          MOVL    #1, R0                      ; SS$_NORMAL
                    04  0077          RET
                        0078
                        0078     ;+
                        0078     ;        Error if event flag number reserved to system or
                        0078     ;        out of range.
                        0078     ;-
                        0078
                        0078     RES_SYS_1:
   50    00000000'8F D0 0078          MOVL    #LIB$_EF_RESSYS, R0         ; Event flag reserved
                    04  007F          RET
                        0080
                        0080     ;+
                        0080     ;        Error if event flag already free.
                        0080     ;-
                        0080
                        0080     ALR_FRE:
   50    00000000'8F D0 0080          MOVL    #LIB$_EF_ALRFRE, R0         ; Event flag already free
                    04  0087          RET
                        0088
```

```
      0088          .SBTTL  LIB$RESERVE_EF - Reserve a local event flag
      0088  ;++
      0088  ; FUNCTIONAL DESCRIPTION:
      0088  ;
      0088  ;       LIB$RESERVE_EF is used when a routine wants to allocate
      0088  ;       a particular local event flag number.  This is different
      0088  ;       from LIB$GET_EF which allocates an arbitrary event flag.
      0088  ;
      0088  ;       To deallocate an event flag reserved with LIB$RESERVE_EF
      0088  ;       use LIB$FREE_EF.
      0088  ;
      0088  ; CALLING SEQUENCE:
      0088  ;
      0088  ;       status.wlc.v = LIB$RESERVE_EF (ef_number.rl.r)
      0088  ;
      0088  ; INPUT PARAMETERS:
      0088  ;
      0088  ;       ef_number       - The number of the event flag desired to be
      0088  ;                         allocated.
      0088  ;
      0088  ; IMPLICIT INPUTS:
      0088  ;
      0088  ;       EF_POOL, a table of available event flags located in OWN
      0088  ;       storage.
      0088  ;
      0088  ; OUTPUT PARAMETERS:
      0088  ;
      0088  ;       NONE
      0088  ;
      0088  ; IMPLICIT OUTPUTS:
      0088  ;
      0088  ;       An entry is made in EF_POOL indicating that the event
      0088  ;       flag is allocated.
      0088  ;
      0088  ; FUNCTION VALUE:
      0088  ; COMPLETION CODES:
      0088  ;
      0088  ;       SS$_NORMAL      - Routine successfully completed.
      0088  ;
      0088  ;       LIB$_EF_ALRRES  - Event flag already reserved.
      0088  ;
      0088  ;       LIB$_EF_RESSYS  - Event flag reserved to system.  This
      0088  ;                         occurs if ef_number is outside the ranges
      0088  ;                         of 1-23 and 32-63.
      0088  ;
      0088  ; SIDE EFFECTS:
      0088  ;
      0088  ;       NONE
      0088  ;
      0088  ;--
      0088
4000  0088          .ENTRY LIB$RESERVE_EF, ^M<IV>   ; Save nothing
      008A
      008A  ;+
      008A  ;       First check to see if ef_number is in range.
      008A  ;-
      008A
```

```
     3F    04 BC    D1  008A          CMPL    @ef_number(AP), #63      ; Greater than 63?
              31    14  008E          BGTR    RES_SYS_2                ; Yes, error
           04 BC    D5  0090          TSTL    @ef_number(AP)           ; Less than 1?
              2C    15  0093          BLEQ    RES_SYS_2                ; Yes, error
     17    04 BC    91  0095          CMPB    @ef_number(AP), #23      ; 1-23?
              15    15  0099          BLEQ    OK_LOW2                  ; Yes, ok
     20    04 BC    91  009B          CMPB    @ef_number(AP), #32      ; 32-63?
              20    19  009F          BLSS    RES_SYS_2                ; No, error
                        00A1
                        00A1   ;+
                        00A1   ;      Now attempt to allocate an event flag in the 32-63 range.
                        00A1   ;-
                        00A1
                        00A1   OK_2:
  50     3F    04 BC    C3  00A1      SUBL3   @ef_number(AP), #63, R0 ; Convert to bit offset
  1B 00000000'EF    50  E2  00A6      BBSS    R0, EF_POOL, ALR_RES    ; Reserve it, but error
                        00AE                                          ; if already reserved.
              0D    11  00AE          BRB     LEAVE_RES
                        00B0
                        00B0   ;+
                        00B0   ;      Now attempt to allocate an event flag in the 1-23 range.
                        00B0   ;-
                        00B0
                        00B0   OK_LOW2:
  50     3F    04 BC    C3  00B0      SUBL3   @ef_number(AP), #63, R0 ; Convert to bit offset
  0C 00000000'EF    50  E5  00B5      BBCC    R0, EF_POOL, ALR_RES    ; Reserve it, but error
                        00BD                                          ; if already reserved.
                        00BD
                        00BD   ;+
                        00BD   ;      Return success
                        00BD   ;-
                        00BD
                        00BD   LEAVE_RES:
        50    01    D0  00BD          MOVL    #1, R0                  ; SS$_NORMAL
                    04  00C0          RET                             ; Exit
                        00C1
                        00C1   ;+
                        00C1   ;      Error if event flag number reserved to system or out
                        00C1   ;      of range.
                        00C1   ;-
                        00C1
                        00C1   RES_SYS_2:
  50     00000000'8F    D0  00C1      MOVL    #LIB$_EF_RESSYS, R0     ; Reserved to system
                    04  00C8          RET                             ; Exit
                        00C9
                        00C9   ;+
                        00C9   ;      Error if already allocated.
                        00C9   ;-
                        00C9
                        00C9   ALR_RES:
  50     00000000'8F    D0  00C9      MOVL    #LIB$_EF_ALRRES, R0     ; Already reserved
                    04  00D0          RET                             ; Exit
                        00D1
                        00D1          .END
```

```
ALL_OUT              00000033 R      02
ALR_FRE              00000080 R      02
ALR_RES              000000C9 R      02
EF_NUMBER          = 00000004
EF_POOL              00000000 R      01
FOUND                00000022 R      02
FOUND_LOW            00000018 R      02
LEAVE_FRE            00000074 R      02
LEAVE_GET            0000002A R      02
LEAVE_RES            000000BD R      02
LIB$FREE_EF          0000003F RG     02
LIB$GET_EF           00000000 RG     02
LIB$RESERVE_EF       00000088 RG     02
LIB$_EF_ALRFRE       ********    X   00
LIB$_EF_ALRRES       ********    X   00
LIB$_EF_RESSYS       ********    X   00
LIB$_INSEF           ********    X   00
OK_1                 00000058 R      02
OK_2                 000000A1 R      02
OK_LOW               00000067 R      02
OK_LOW2              000000B0 R      02
RES_SYS_1            00000078 R      02
RES_SYS_2            000000C1 R      02
SCAN                 00000002 R      02

                                        +----------------+
                                        ! 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$DATA                       00000008  (     8.)  01 (  1.)    PIC   USR   CON   REL   LCL NOSHR NOEXE   RD    WRT NOVEC LONG
_LIB$CODE                       000000D1  (   209.)  02 (  2.)    PIC   USR   CON   REL   LCL   SHR   EXE   RD  NOWRT NOVEC LONG

                                     +------------------------+
                                     ! Performance indicators !
                                     +------------------------+

Phase                      Page faults    CPU Time       Elapsed Time
-----                      -----------    --------       ------------
Initialization                      32    00:00:00.03    00:00:02.00
Command processing                 125    00:00:00.32    00:00:02.10
Pass 1                              73    00:00:00.52    00:00:03.20
Symbol table sort                    0    00:00:00.00    00:00:00.00
Pass 2                              81    00:00:00.49    00:00:03.37
Symbol table output                  3    00:00:00.04    00:00:00.47
Psect synopsis output                3    00:00:00.01    00:00:00.01
Cross-reference output               0    00:00:00.00    00:00:00.00
Assembler run totals               319    00:00:01.42    00:00:11.16

The working set limit was 1050 pages.
4949 bytes (10 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 24 non-local and 0 local symbols.
430 source lines were read in Pass 1, producing 19 object records in Pass 2.
0 pages of virtual memory were used to define 0 macros.
```

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

Macro library name                Macros defined
------------------                --------------
_$255$DUA28:[SYSLIB]STARLET.MLB;2         0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

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

LIB$
1-00

LIBEMODH
LIS

LIBEMODD
LIS

LIBESTEMU
LIS

LIBEF
LIS

LIBEMULAT
LIS

LIBFFS
LIS

LIBFINCVT
LIS

LIBE2AREV
LIS

LIBFAO
LIS

LIBEMODG
LIS

LIBEXTV
LIS

LIBEDIV
LIS

LIBESTABL
LIS

LIBFFC
LIS

LIBEMODF
LIS

LIBEMUL
LIS

LIBFILSCA
LIS

LIBEXTZV
LIS

LIBEBCASC
LIS

LIBFAOL
LIS