D 8

```
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**LIBASTINP

```
LL            IIIIII    BBBBBBBB      AAAAAA      SSSSSSSS  TTTTTTTTTT    IIIIII    NN      NN  PPPPPPPP
LL            IIIIII    BBBBBBBB      AAAAAA      SSSSSSSS  TTTTTTTTTT    IIIIII    NN      NN  PPPPPPPP
LL              II      BB      BB  AA      AA  SS              TT          II      NN      NN  PP      PP
LL              II      BB      BB  AA      AA  SS              TT          II      NN      NN  PP      PP
LL              II      BB      BB  AA      AA  SS              TT          II      NNNN    NN  PP      PP
LL              II      BB      BB  AA      AA  SS              TT          II      NNNN    NN  PP      PP
LL              II      BBBBBBBB    AA      AA    SSSSSS        TT          II      NN  NN  NN  PPPPPPPP
LL              II      BBBBBBBB    AA      AA    SSSSSS        TT          II      NN  NN  NN  PPPPPPPP
LL              II      BB      BB  AAAAAAAAAA          SS      TT          II      NN    NNNN  PP               ....
LL              II      BB      BB  AAAAAAAAAA          SS      TT          II      NN    NNNN  PP               ....
LLLLLLLLLL    IIIIII    BBBBBBBB    AA      AA  SSSSSSSS        TT        IIIIII    NN      NN  PP               ....
LLLLLLLLLL    IIIIII    BBBBBBBB    AA      AA  SSSSSSSS        TT        IIIIII    NN      NN  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
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

Table of contents

```
    (2)       52        DECLARATIONS
    (3)       85        LIB$AST_IN_PROG  - is AST in progress?
```

```
0000          .TITLE  LIB$AST_IN_PROG - Inquire if AST in progress
0000          .IDENT  /1-004/                 ; File: LIBASTINP.MAR Edit DG1004
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    ;       Return SUCCESS (1) if an AST is active in the current mode.
0000    ;
0000    ; ENVIRONMENT: All Modes, AST Reentrant
0000    ;
0000    ;--
0000    ; AUTHOR: Jonathan M. Taylor, CREATION DATE: 23-Jan-78
0000    ;
0000    ; MODIFIED BY:
0000    ;
0000    ;       JMT, 23-Jan-78 : VERSION 0
0000    ; 01    - Original
0000    ; 1-001 - Update version number and copyright notice.  The edit number
0000    ;           on version 0 was 4.  JBS 16-NOV-78
0000    ; 1-002 - Add "_" to the PSECT directive.  JBS 21-DEC-78
0000    ; 1-003 - Allocate an event flag number for the $GETJPI using LIB$GET_EF.
0000    ;         SBL 24-Nov-1981
0000    ; 1-004 - Use $GETJPIW to ensure synchronous operation.  DG 26-Oct-1983
```

```
    0000          .SBTTL  DECLARATIONS
    0000
    0000  ;
    0000  ; EXTERNAL SYMBOLS:
    0000  ;
    0000          .DSABL  GBL
    0000          .EXTRN  LIB$STOP
    0000          .EXTRN  LIB$GET_EF
    0000          .EXTRN  LIB$FREE_EF
    0000          .EXTRN  LIB$_FATERRLIB
    0000
    0000  ;
    0000  ; INCLUDE FILES:
    0000  ;
    0000
    0000  ;
    0000  ; MACROS:
    0000  ;
    0000          $JPIDEF                         ; define system service constants
    0000
    0000  ;
    0000  ; EQUATED SYMBOLS:
    0000  ;       none
    0000
    0000  ;
    0000  ; OWN STORAGE:
    0000  ;       none
    0000
    0000  ;
    0000  ; PSECT DECLARATIONS:
    0000  ;
00000000          .PSECT _LIB$CODE PIC, SHR, LONG, EXE, NOWRT
```

```
                  0000    85          .SBTTL  LIB$AST_IN_PROG  - is AST in progress?
                  0000    86
                  0000    87  ;++
                  0000    88  ; FUNCTIONAL DESCRIPTION:
                  0000    89  ;
                  0000    90  ;       Call system service GETJPI to get access modes with active
                  0000    91  ;       ASTs.  Pick off the bit corresponding to the current-mode
                  0000    92  ;       (taken from the PSL) and return it.
                  0000    93  ;
                  0000    94  ; CALLING SEQUENCE:
                  0000    95  ;
                  0000    96  ;       at_ast_level.wl.v = LIB$AST_IN_PROG ([efn.rl.r])
                  0000    97  ;
                  0000    98  ; INPUT PARAMETERS:
                  0000    99  ;
     00000004     0000   100          efn = 4                 ; Longword event flag number to use for $GETJPI,
                  0000   101                                  ; passed by reference.  Optional.  If omitted,
                  0000   102                                  ; LIB$GET_EF is called to allocate a number.
                  0000   103  ;
                  0000   104  ; IMPLICIT INPUTS:
                  0000   105  ;
                  0000   106  ;       NONE
                  0000   107  ;
                  0000   108  ; OUTPUT PARAMETERS:
                  0000   109  ;
                  0000   110  ;       NONE
                  0000   111  ;
                  0000   112  ; IMPLICIT OUTPUTS:
                  0000   113  ;
                  0000   114  ;       NONE
                  0000   115  ;
                  0000   116  ; FUNCTION VALUE:
                  0000   117  ;
                  0000   118  ;       Returns SUCCESS (1) if we're currently as AST level, otherwise 0.
                  0000   119  ;
                  0000   120  ;
                  0000   121  ; SIDE EFFECTS:
                  0000   122  ;
                  0000   123  ;       May signal LIB$_FATERRLIB
                  0000   124  ;
                  0000   125  ;--
            0000  0000   126          .ENTRY  LIB$AST_IN_PROG, ^M<>
         7E   D4  0002   127          CLRL    -(SP)                           ; set up a one-item item-list
         7E   7C  0004   128          CLRQ    -(SP)                           ;        +---------------+
      08 AE   DF  0006   129          PUSHAL  8(SP)                           ;ITMLST: |_0300_|___4___|
03000004 8F   DD  0009   130          PUSHL   #<JPI$_ASTACT@16>+4             ;        |-------*-------+---+
                  000F   131                                                  ;        |-------0-------|   |
                  000F   132                                                  ;        |-------0-------|   |
                  000F   133                                                  ;        |-------0-------| <-+
   51    5E   D0  000F   134          MOVL    SP, R1
         6C   95  0012   135          TSTB    (AP)                            ; Zero arguments?
         1E   13  0014   136          BEQL    GET_EF                          ; Yes, call LIB$GET_EF
      04 AC   D5  0016   137          TSTL    efn(AP)                         ; Omitted by reference?
         19   13  0019   138          BEQL    GET_EF                          ; Yes, call LIB$GET_EF
                  001B   139          $GETJPIW_S      EFN=@efn(AP), ITMLST=(R1)
      4D 50   E9  002F   140          BLBC    R0, ERROR                       ; Signal if error
         35   11  0032   141          BRB     OK                              ; Skip to common processing
```

```
                              0034
                              0034   ;+
                              0034   ; Come here if we need to allocate an event flag number
                              0034   ;-
                              0034   GET_EF:
                  7E     D4   0034            CLRL     -(SP)                         ; Make a place for the event flag number
                  6E     9F   0036            PUSHAB   (SP)                          ; Address of efn
      00000000'GF 01     FB   0038            CALLS    #1, G^LIB$GET_EF              ; Get an efn
               3D 50     E9   003F            BLBC     R0, ERROR                     ; Exit if unsuccessful
            50    5E     D0   0042            MOVL     SP, R0                        ; Move address of EFN
                              0045            $GETJPIW_S      EFN=(R0), ITMLST=(R1)
               24 50     E9   0058            BLBC     R0, ERROR                     ; branch if not successful
                  6E     9F   005B            PUSHAB   (SP)                          ; Address of efn
      00000000'GF 01     FB   005D            CALLS    #1, G^LIB$FREE_EF             ; Free the efn
               18 50     E9   0064            BLBC     R0, ERROR                     ; Signal if error
                  00     11   0067            BRB      OK                            ; Skip to finish up
                              0069
                              0069   ;+
                              0069   ; Come here for final processing
                              0069   OK:
         50    FC AD     D0   0069            MOVL     -4(FP), R0                    ; R0 = bitvector by mode
                  51     DC   006D            MOVPSL   R1
         51    51 08     9C   006F            ROTL     #8, R1, R1
         51    FC 8F     8A   0073            BICB     #-4, R1                       ; R1<0:7> = current mode
            51    51     8E   0077            MNEGB    R1, R1                        ; R1 = 0,-1,-2,-3
         50    50 51     9C   007A            ROTL     R1, R0, R0                    ; get proper flag into lsb
                  04          007E            RET
                              007F
                              007F   ERROR:
                  50     DD   007F            PUSHL    R0                            ; Push original error code
                  7E     D4   0081            CLRL     -(SP)                         ; Zero FAO parameters
         00000000'8F     DD   0083            PUSHL    #LIB$_FATERRLIB
      00000000'GF 03     FB   0089            CALLS    #3, G^LIB$STOP                ; Can't return
                  04          0090            RET                                    ; But have signalled PC in this module
                              0091
                              0091            .END
```

```
$$T1                             = 00000001
EFN                              = 00000004
ERROR                              0000007F R       02
GET_EF                             00000034 R       02
JPI$_ASTACT                      = 00000300
LIB$AST_IN_PROG                    00000000 RG      02
LIB$FREE_EF                        ********    X    00
LIB$GET_EF                         ********    X    00
LIB$STOP                           ********    X    00
LIB$_FATERRLIB                     ********    X    00
OK                                 00000069 R       02
SYS$GETJPIW                        ********   G     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
$ABS$                              00000000  (      0.)  01 (  1.)  NOPIC    USR    CON    ABS    LCL NOSHR   EXE   RD     WRT NOVEC BYTE
_LIB$CODE                          00000091  (    145.)  02 (  2.)    PIC    USR    CON    REL    LCL   SHR   EXE   RD   NOWRT NOVEC LONG

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

Phase                        Page faults    CPU Time       Elapsed Time
-----                        -----------    --------       ------------
Initialization                        29    00:00:00.05    00:00:00.81
Command processing                   113    00:00:00.28    00:00:03.53
Pass 1                               133    00:00:01.19    00:00:07.03
Symbol table sort                      0    00:00:00.08    00:00:00.35
Pass 2                                47    00:00:00.36    00:00:01.97
Symbol table output                    3    00:00:00.01    00:00:00.01
Psect synopsis output                  2    00:00:00.01    00:00:00.01
Cross-reference output                 0    00:00:00.00    00:00:00.00
Assembler run totals                 329    00:00:01.99    00:00:13.72

The working set limit was 900 pages.
8430 bytes (17 pages) of virtual memory were used to buffer the intermediate code.
There were 10 pages of symbol table space allocated to hold 120 non-local and 0 local symbols.
177 source lines were read in Pass 1, producing 13 object records in Pass 2.
11 pages of virtual memory were used to define 10 macros.

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

Macro library name                               Macros defined
------------------                               --------------
_$255$DUA28:[SYSLIB]STARLET.MLB;2                       7

200 GETS were required to define 7 macros.

There were no errors, warnings or information messages.
```

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

0203 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

RTLMACMAR
MAR

LIBASCEBC
LIS

LIBASTINP
LIS

RTLMACB32
REQ

STRMACROS
REQ

LIBBINTRE
LIS

RTLODBG
REQ

LIBDEF
FOR

LIBANASTR
LIS

LIBASNMBX
LIS

LIBBBCCI
LIS

LIBCHAR
LIS

LIBABUPCA
LIS

LIBADDX
LIS

RTLPSECT
REQ

SIGDEF
FOR

LIBASCTIM
LIS

LIBTABMAC
MAR

LIBATTACH
LIS

STRLNK
REQ.

LIBA2EREV
LIS

LIBBBSSI
LIS

LIBCALLG
LIS

LIBCLICAL
LIS

LIBADDP
LIS