```
111111111
                                                                   TTTTTTTTTTTTT
                    TITITITITITI
                                                                                   LLL
                    LLL
                                                                   TTTTTTTTTTTTT
                                                                                   LLL
                                             888
888
888
888
                                 888
                                                  RRR
LLL
                       III
                                                              RRR
                                                                         TTT
                                                                                    LLL
                       III
                                 888
                                                  RRR
                                                              RRR
LLL
                                                                         TIT
                                                                                    LLL
                                 888
888
                                                  RRR
                                                              RRR
                       H
LLL
                                                                         TTT
                                                                                    LLL
                                                  RRR
                                                              RRR
                       III
LLL
                                                                         TIT
                                                                                    LLL
                                 888
                                             BBB
                                                              RRR
                                                  RRR
                       III
LLL
                                                                         TTT
                                                                                    LLL
                                 BBB
                                             BBB
                       III
                                                  RRR
                                                              RRR
LLL
                                                                         TIT
                                                                                    LLL
                                 III
                                                  RRRRRRRRRRR
LLL
                                                                         TTT
                                                                                    LLL
                                                  RRRRRRRRRRRR
LLL
                       111
                                                                         TIT
                                                                                    LLL
                                 88888888888
                                                  RRRRRRRRRRRR
LLL
                       111
                                                                         TIT
                                                                                    LLL
                                 888
                                                  RRR
                                                        RRR
                                             BBB
LLL
                       111
                                                                         TTT
                                                                                    LLL
                                 BBB
                                             BBB
                                                  RRR
                                                        RRR
                       111
LLL
                                                                         TIT
                                                                                    LLL
                       ĬĬĬ
                                 888
                                                  RRR
                                                        RRR
LLL
                                             BBB
                                                                         TTT
                                                                                    LLL
                       III
                                 888
                                             BBB
                                                  RRR
LLL
                                                           RRR
                                                                         TTT
                                                                                    LLL
                       III
                                 888
                                             BBB
                                                  RRR
LLL
                                                           RRR
                                                                         TTT
                                                                                    LLL
LLL
                       111
                                 BBB
                                             BBB
                                                  RRR
                                                           RRR
                                                                         TIT
                                                                                    LLL
                                 LLLLLLLLLLLLLLL
                    1111111111
                                                  RRR
                                                              RRR
                                                                         TTT
                                                                                    LLLLLLLLLLLLL
LLLLLLLLLLLLLL
                    RRR
                                                              RRR
                                                                         TTT
                                                                                   LLLLLLLLLLLLLL
RRR
                                                              RRR
                    111111111
                                                                         III
                                                                                   LLLLLLLLLLLLLL
```

1

Sy

LI LI LI

000000 000000 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00		\$		BBBBBBBB BBBBBBBB BB BB BB BB BB BB BBBBBB	••••
\$	DDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDDD				

•

```
LIB
```

i s

Mnf

! F

MAC

```
{ REQUIRE file for Logical Unit Block (LUB)
{ File: OTSLUB.SDL Edit: MDL2005
     COPYRIGHT (c) 1978, 1980, 1982, 1984 BY DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
      ALL RIGHTS RESERVED.
*****
     THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
      OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
      TRANSFERRED.
     THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
      AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
      CORPORATION.
      DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
      SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
(*
   Author: T.Hastings
   Change history:
 [Previous edit history removed. SBL 24-Aug-1982]
1-048 - Give LUB$A_ASSOC_VAR another name LUB$L_ALQ, this is to hold the allocation quantity for files in BASIC. FM 1-Oct-1980
  1-049 - Add a flag to request ANSI processing. PLL 22-Jul-1982 2-001 - Convert to SDL. SBL 24-Aug-1982
{ 2-002 - Don't depend on names for sub-structures. SBL 29-Sep-1982 { 2-003 - Change aggregate name to LUB for better fieldset name. SBL 26-Oct-1982 { 2-004 - Add fields for RFA cacheing. SBL 2-June-1983 { 2-005 - add field to indicate FIELDing. MDL 29-Mar-1984
{ The LUB contains OTS OWN data associated with a
{ particular logical unit which is needed between I/O statements.
   OWN data which is needed for several calls within a single
{ I/O statement is allocated in the I/O Statement Block (ISB).
   Data which is needed during a single call is always LOCAL.
{ LUB definition (-11 OTS equivalents are indicated in parens)
( Symbols are of the form: LUB$t_symbol where t is
{ A,B,W,L,T,or V.
MODULE SLUBDEF:
AGGREGATE LUB STRUCTURE PREFIX LUBS ORIGIN end_of_lub;
EDefine some constants that are used to set the organization field of the LUB
```

```
OTSLUB.SDL:1
{-
CONSTANT (
    ORG_SEQUE,
ORG_RELAT,
ORG_INDEX,
ORG_TERMI,
ORG_VIRTU
                  { Organization sequential { Organization relative
                    Organization indexed sequential
                  Corganization virtual
                    Organization terminal format
    ) EQUALS 1 INCREMENT 1;
{+
{ Define the symbols for the special LUNs with negative numbers.
CONSTANT (
    LUN_BPRI.
                  { Logical unit for BASIC PRINT { Logical unit no. for BASIC INPUT
    LUN_INPU,
    LUN BREAD,
                  { Logical unit for BASIC READ { Logical unit for FORTRAN ENCODE/DECODE
    LUN ENCD.
    LUN_READ,
                    Logical unit no. for FORTRAN READ
    LUN_ACCE,
                  { Logical unit no. for FORTRAN TYPE { Logical unit no. for FORTRAN PRINT
    LUN TYPE,
    LUNTPRIN
    ) EQUALS -8 INCREMENT 1:
                           EQUALS LUB$K_LUN_BPRI;
EQUALS LUB$K_LUN_READ;
EQUALS LUB$K_LUN_PRIN;
CONSTANT ILUN MIN
                                                      { Min LUN (for LUN table)
CONSTANT DLUN MIN CONSTANT DLUN MAX
                                                      { Min default-OPEN LUN for FORTRAN
                                                      { Max default-OPEN LUN for FORTRAN
CONSTANT LUN MIN
                           EQUALS 0:
                                                      { Min. explicit FORTRAN OPEN LUN
                                                        BASIC limit is .GT. this.
CONSTANT LUN MAX
                           EQUALS 119:
                                                      { Max. explicit or implicit LUN
{ Define a symbol for the default value of the right margin. This value
{ is used only by BASIC. {-
CONSTANT D_MARGIN
                           EQUALS 72:
                                                      { default right margin for files
{ Define a symbol for the maximum length of the prompt buffer.
{ This much space is allocated whenever a sequential file is opened
( in case we are to prompt for input from it.
CONSTANT PBUF_SIZ
                           EQUALS 80:
                                                      { Size of prompt buffer
CONSTANT NEG_BLN EQUALS . ;
                                             { define negative length of LUB
                                             { RMS User Buffer.
    UBF ADDRESS:
     union 1 UNION:
         UNIT_STT3 WORD UNSIGNED;
                                             { more flags
         UNIT_STT3_STRUCT STRUCTURE;
```

LIB

MAC

MAC

! E

```
**
```

```
NOECHO BITFIELD:
                                                                     If the BASIC function NOECHO has been done.
                                                                      (applies to terminals only)
If the BASIC function ONECHR has been done
                     ONECHR BITFIELD:
                                                                     (applies to terminals only)
Cancel control O (BASIC function RCTRLO)
                     CCO BITFIELD:
                                                                      (applies to terminals only)
                                                                      1 if last direct operation was FIND
                     FIND_LAST BITFIELD:
                                                                      (FORTRAN direct access)
                     PTA BITFIELD:
                                                                     Purge type ahead
                                                                      (applies to terminals only)
(BAS-new) Used in I/O element transmit
                     AST_GUARD BITFIELD:
                                                                     to detect concurrence of ASTs using the I/O data base. If concurrence is detected then the routine is repeated.
                                                                    then the routine is repeated.

1 if FAB$V_(R is set. (FORTRAN)

1 if FAB$V_FTN is set. (FORTRAN)

(BAS-new) =1, PRN format and semantics for Basic stream files.

1 if FAB$V_PRN is set. (FORTRAN)

(BAS-new) =1, the right margin for terminal format files is infinite.

1 if file opened with USEROPEN

1 if file is not sequential organization
                     CR BITFIELD:
                    FIN BITFIELD;
                    . . N BITFIELD;
                    NOMARGIN BITFIELD:
                    USEROPEN BITFIELD:
                                                                    1 if file is not sequential organization. Used by FORTRAN ENDFILE.
1 if ANSI INPUT
                    NOTSEGORG BITFIELD:
                    ANSI BITFIELD:
                    RFA_CACHE_ENABLE BITFIELD; { 1 if RF/
FIELD_USE BITFIELD; { 1 if FIE
fill_T BITFIELD LENGTH 1 FILL TAG $$;
END_UNIT_STT3_STRUCT;
                                                                     1 if RFA cacheing enabled (FORTRAN)
                                                                     1 if FIELD stmt on this ch. (BASIC)
TAG $$; { This many bits remain in this status word
              END union_1;
       BLS WORD UNSIGNED:
                                                                 i Mag tape block size, from FAB$W_BLS
{+
{ The following address, if non-zero, points to a routine to be called
{ just before the LUB is CLOSEd. This is used by the BASIC File Array
{ support to write out the last buffer.
       CLOSE ADDRESS:
                                                                  { Call here on CLOSE
{+
{ The following quadword is used to link the LUB to the LUB table
{ maintained by OTSCCB.
{-
       QUEUE QUADWORD UNSIGNED:
                                                                  { Link for INSQUE and REMQUE instructions
LUB Locations used by all User Data Formatted (UDF) level Procedures
{ which are: FOR$$UDF_{R,W}{F,U,L} and BAS$$UDF_{RW}_L {-
       BUF_PTR ADDRESS:
                                                                  { (FOR-BLBUF) Adr. of next byte in buffer to be
                                                                  { filled or emptied with user data { (FOR-EOLBUF) Adr.+1 of last byte in buffer
       BUF_END ADDRESS:
```

```
E 14
16-SEP-1984 16:41:41.68 Page
OTSLUB.SDL:1
                                              { to be filled or emptied with user data.
{+ {
    This is the buddy pointer for BASIC I/O. For all units except 0, it
{ should point to itself. For unit 0, the Print CCB will point to the
{ Input CCB and vice versa
    BUDDY PTR ADDRESS:
                                               { pointer to the complementary CCB for Print and
                                              { Input. Needed for recursive and continued I/O
  LUB Locations used solely by the input or output dependent
{ Formatted User Data Formatters .
    BUF_BEG ADDRESS:

    ⟨ (FOR-LNBUF) Adr. of first byte in buffer (FOR-used)

                                                 for T format).
                                                 (FOR-TSPECP) Adr. of highest byte filled in
    BUF_HIGH ADDRESS:
                                                 buffer during format processing (FOR-needed
                                               { because In format can move backwards).
  LUB locations set by OPEN, default OPEN, CALL FDBSET, or
[ DEFINE FILE, and checked on every 1/0 statement
     ORGAN BYTE UNSIGNED:
                                                 (BAS-If.BLK-If.TRF) file organization:
                                                 virtual block, sequential, relative, indexed
                                                 sequential or terminal format.
                                                Bucket size, from FAB$B_BKS
     BKS BYTE UNSIGNED:
                                                 Logical unit number (0:99)
    LUN WORD:
                                              { Note: signed! Negative LUNS used for: { INPUT and PRINT
    union 1A UNION FILL:
         PRINT_POS LONGWORD UNSIGNED:
                                                 (BAS-POSITN) printhead position
                                                 PRINT statements may end in a semicolon or
                                                a comma requiring the printhead position to be maintained to the next PRINT statement.
                                                This is a longword because the longest string
                                                 (65K) may be put in the longest buffer (65K).
    RFA_CACHE_BEG ADDRESS;
END union_1A;
union_1B_UNION_FILL;
                                               { First allocated byte of RFA cache (FOR)
    WAIT TIME LONGWORD UNSIGNED;
RFA_CACHE_PTR ADDRESS;
END_union_1B;
IFI WORD UNSIGNED;
                                               { (BAS-WATIM) Wait time for a WAIT operation
                                               { Pointer to current RFA cache entry (FOR)
                                                 RMS internal file id, needed
                                                 until $CLOSE
     RBUF_SIZE WORD UNSIGNED;
                                                 Record buffer size in bytes
                                                 Set by OPEN, default open, or DEFINE FILE.
                                                 Used to allocate record buffer at open.
                                                 Read by record level of abstraction FORSSREC_{R,W}(F,U,L)
                                              ((BAS-new) the right margin for 
(a terminal format file. The default is 
(72 for terminal format files, set to terminal
     R_MARGIN WORD UNSIGNED;
```

LIE

į p

F

```
F 14
16-SEP-1984 16:41:41.68 Page
OTSLUB.SDL:1
                                               width otherwise. Not the same as
                                               buffer size because of embedded carriage con-
                                               trol characters. For terminal format files,
                                               when the cursor position exceeds this value
                                               the record is PUT.
    D_MARGIN WORD UNSIGNED:
                                             { Default right margin.
    LANGUAGE BYTE UNSIGNED:
                                               The language that opened the LUN, as follows:
    CONSTANT LANG MIN CONSTANT LANG NONE
                           EQUALS 0:
                                               Minimum language code
                           EQUALS 0:
                                               None (probably not open yet)
    CONSTANT LANG BAS
CONSTANT LANG FOR
CONSTANT LANG MAX
                           EQUALS 1:
                                               VAX-11 BASIC-PLUS-2
                           EQUALS 2:
                                               VAX-11 FORTRAN-IV PLUS
                           EQUALS 2:
                                             { Maximum language code
    RFM BYTE UNSIGNED:
                                             Record format, from FAB$B_RFM
    union_2 UNION;
        BAS_VFC WORD UNSIGNED;
                                               (BAS-new) fixed control block for carriage control
                                               This is pointed to by the RAB so it is a part
                                               of the RMS interface. As a result, it is only
                                             { written to by the REC level.
        BAS_VFC_STRUCT STRUCTURE:
             BAS_VFC1 BYTE UNSIGNED;
BAS_VFC2 BYTE UNSIGNED;
                                             { overlay first byte of VFC
                                             { overlay second byte of VFC
             END BAS_VFC_STRUCT;
        END union_2;
    union 3 UNION:
         ASSOC_VAR ADDRESS:
                                             { Adr. of ASSOCIATEVARIABLE or 0 if none
                                               Set by OPEN or DEFINEFILE.
                                             { LUB$V_ASS_VAR_L specifies word/longword
         ALQ LONGWORD UNSIGNED:
                                             { Allocation quantity for files, also used in assoc_var
        END union_3;
    LOG_RECNO LONGWORD UNSIGNED;
                                               current or next Logical (or segmented)
                                               record number for sequential access files
                                               (needed for BACKSPACE of segmented records). Current or next
                                               record number for
                                               FORTRAN direct access files (0=1=first record)
    REC_MAX LONGWORD UNSIGNED;
                                               Direct access maximum record number
                                               (RMS doesn't keep for Sequential organization files). Set by DEFINE FILE or OPEN.
Address of FAB allocated by CALL ASSIGN, CALL FDBSET, DEFINEFILE or OPEN.

O = ASSIGN, FDBSET, DEFINEFILE or
    FAB ADDRESS:
                                               LUB not done.
                                               NOTE: This field cannot move from an offset
                                               of -24 due to Fortran compatibility routines
    RBUF_ADR ADDRESS:
                                               Size in bytes of record buffer (includes
                                               any FORTRAN information kept in the record)
    DID WORD UNSIGNED DIMENSION 3;
                                               three words to hold directory ID from OPEN
    RAT BYTE UNSIGNED;
                                               The record attributes, from FAB$B_RAT.
    RSL BYTE UNSIGNED:
                                               size of resultant name string (0 = no string allocated)
    RSN ADDRESS:
                                               address of resultant name string
```

union\_4 UNION;

LIE

i s

SWI

271

LIE

!+

**PSE** 

! E

```
UNIT_ATTR WORD UNSIGNED:
                                                       (FOR-D.STAT) Unit attribute bits which are needed between I/O statements.
                                                       NOTE: Some of these bits are in fixed
                                                      positions as noted.
UNIT_ATTR_STRUCT STRUCTURE;
                                                      (FOR-DV.OPN) LUB has been successfully opened by OPEN or default OPEN.
      OPENED BITFIELD:
                                                      Cleared by CLOSE or error during OPEN NOTE: cannot be moved from offset -4,0 due
                                                      to Fortran compatibility.

(FOR--) An I/O statement is active on this logical unit. Set to 0 on an error or end of I/O list. Used to prevent recursive I/O on the same logical unit.
      IO_ACTIVE BITFIELD:
                                                       (FOR-DV.RDO) No writes will (can) be
      READ_ONLY BITFIELD:
                                                       done to this file.
                                                       Set by CALL FDBSET or OPEN 'READONLY'.
NOTE: cannot be moved from offset -4,2 due
                                                       to fortran compatibility.
                                                      (FOR-DV.OLD) Old (existing) file required, do OPEN not CREATE. Set by TYPE='OLD' or FDBSET 'OLD'.
      OLD_FILE BITFIELD:
                                                       NOTE: cannot be moved from offset -4.3 due to
                                                       Fortran compatibility.
                                                       (FOR-DV.9FD) FORTRAN direct access file. Set by ACCESS='DIRECT' or DEFINEFILE.
      DIRECT BITFIELD:
                                                       Note: this bit is independent of RMS
                                                       file organization (Sequential or Relative).
                                                      Can not be moved from -4.4 unless FOR$$IO_BEG is modified. (FOR-DV.SCR) TYPE='SCRATCH' specified.
      SCRATCH BITFIELD:
                                                       (FOR-DV.DEL) OPEN DISP='DELETE' specified.
      DELETE BITFIELD:
                                                       Checked at CLOSE
                                                       (FOR-DV.SPL) OPEN DISPOSE='PRINT' causes
      PRINT BITFIELD:
                                                       spooling at CLOSE. (FOR-DV.FMP) File is FORTRAN formatted.
      FORMATTED BITFIELD:
                                                       OPEN FORM='FORMATTED'
                                                       0 = unspecified.
                                                      NOTE: Can not be moved from -4,8 unless FOR$$10_BEG is modified.
(FOR-DV.UFP) File is FORTRAN unformatted.
0 = unspecified. Set by DEFINE FILE or OPEN.
Note: LUB$V_FORMATTED and LUB$V_UNFORMAT can both be 0 on default OPEN done for
      UNFORMAT BITFIELD:
                                                      END FILE since the format may be either. Can not be moved from -4.9 unless
                                                      fOR$$IO_BEG is modified.
(FOR--) 1 = Record format is RMS fixed (FLR).
      FIXED BITFIELD:
                                                       OPEN RECORDTYPE='FIXED'
                                                      O = Record format is RMS variable
(VLR or VLRM, i.e., VLR on Relative
Organization file. OPEN RECORDTYPE='VARIABLE'
(FDR--) Segmented (unformatted) records are
      SEGMENTED BITFIELD:
                                                       to be used. Otherwise only one
                                                       record (VLR or FLR) is to be read or
```

```
H 14
16-SEP-1984 16:41:41.68 Page
OTSLUB.SDL:1
                                                     written with no segmented control info.
RECORDIYPE = 'SEGMENTED' in OPEN or TYPE not specified
                                                      for sequential unformatted file.
               ASS_VAR_L BITFIELD:
                                                      (FOR-DV.A14) ASSOCIATEVARIABLE is a longword
                                                      0 = ASSOCIATEVARIABLE is a word or not
                                                      present. Set by OPEN or DEFINE FILE. See LUBSA_ASSOC_VAR (FOR-DV.APD) File was opened ACCESS = 'APPEND'
               APPEND BITFIELD:
                                                      also used as state bit (LOG RECNO is undefined)
                                                      NOTE: cannot be moved from offset -4,13 due to
                                                      Fortran compatibilty.
1 if ACCESS='SEQUENTIAL' (FORTRAN)
               SEQUENTIA BITFIELD:
                                                     Note: Can not be moved from -4.14 unless FOR$$IO BEG is modified.
1 if ACCESS='KEYED' (FORTRAN)
               KEYED BITFIELD:
                                                      Note: Can not be moved from -4,15
                                                    { unless FOR$$IO BEG is modified.
               END UNIT_ATTR_STRUCT;
          END union 4:
Bits set by OPEN, default OPEN, CALL FDBSET, or
( DEFINEFILE, and checked on every I/O statement
     union_5 UNION;
UNIT_STT2 WORD UNSIGNED;
UNIT_STT2_STRUCT_STRUCTURE;
                                                   { Second word of bits
               VIRT_RSN BITFIELD;
                                                     indicates that RSN points to dynamic memory
                                                      not local storage
               ENDFILOPN BITFIELD:
                                                     File was implicitly opened to do ENDFILE
                                                      When first I/O is done, there are a few
                                                      defaults which will be specified:
                                                     LUBSV_FORMATTED or LUBSV_UNFORMAT
LUBSV_SEGMENTED
                                                     Then EUB$V ENDFIL OPN is cleared. (BAS-new) The last output element transmitter ended
               FORM_CHAR BITFIELD;
                                                      in a comma or semicolon.
                                                      (BAS-IF.WRT) the PRINT buffer already has
               OUTBUF_DR BITFIELD;
                                                      something in it and should be dumped
                                                      before continuing. Set by BAS$$DD_WRITE
                                                     Used for same purpose by BASIC file Array support (BAS-IF.TRF) terminal format file on any unit including 0
               TERM_FOR BITFIELD:
                                                      Set by OPEN info from user. = 1, term-
                                                      inal format
               TERM_DEV BITFIELD;
                                                      (BAS-IF.TRM) terminal device on any unit including 0
                                                      Set by OPEN info from RMS. = 1, term-
                                                      inal device
                                                      (BAS-IF.FRC) forcible device on any unit including 0
               FORCIBLE BITFIELD:
                                                      Set by OPEN info from RMS. = 1, term-
                                                     inal or line printer device (BAS-new?) terminal device - unit 0 Set by Default OPEN for PRINT and INPUT
               UNIT_O BITFIELD;
                                                      (BAS-IF.VIR) marks a files first usage as a
               VA_USE BITFIELD;
                                                     virtual array. Once used as virtual, it cannot be used for block I/O. (BAS-BIO) Marks a file's first use as block
               BLK_USE BITFIELD;
```

019

!\*1

<u>i</u>-

...

•

...

1

į.

\*

LII

! +

```
015
```

```
I 14
16-SEP-1984 16:41:41.68 Page 8
OTSLUB.SDL:1
                                                                                    { I/O. Once used as block I/O , it cannot be { used for virtual I/O. { (BAS-IF.CON) File is multistream connected. { File is (or was) connected to.
                         M_STREAM BITFIELD;
M_STR_C BITFIELD;
        The following bit is set by CLOSE to indicate that the LUB should be { deallocated as soon as all recursive or nested I/O on it has { completed. It is cleared (in effect) by OTS$$POP_CCB deallocating the { storage. While it is set the LUN may not be OPENed since there is { I/O outstanding which should be allowed to fail. {-
                                                                                       Can deallocate this LUB
FORTRAN DISP='SUBMIT' if set.
FORTRAN BLANK='ZERO' if clear,
BLANK='NULL' if set.
If 1, the record buffer was allocated
by the user don't deallocate it at
                          DEALLOC BITFIELD:
                          SUBMIT BITFIELD;
                         NULLBLNK BITFIELD;
                         USER_RBUF BITFIELD:
                                                                                    { by the user, don't deallocate it at
{ CLOSE time.
                         END UNIT_STT2_STRUCT;
                 END union 5;
         CONSTANT LUB_LEN EQUALS ::
                                                                                    { Length of LUB
        end_of_lub BYTE FILL TAG $$;
        END LUB;
END_MODULE $LUBDEF;
{ End of file OTSLUB.SDL
```

1.

FIE

STF

+ 1 C 0202 AH-BT13A-SE

DIGITAL EQUIPMENT CORPORATION CONFIDENTIAL AND PROPRIETARY

