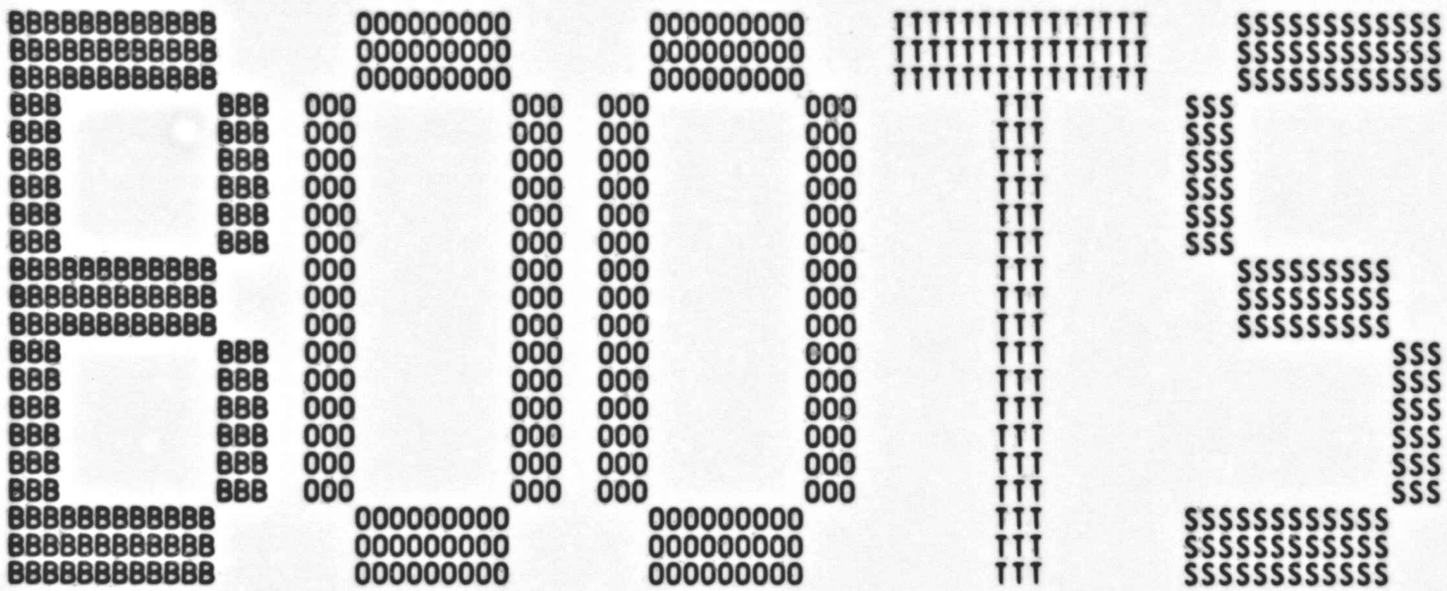
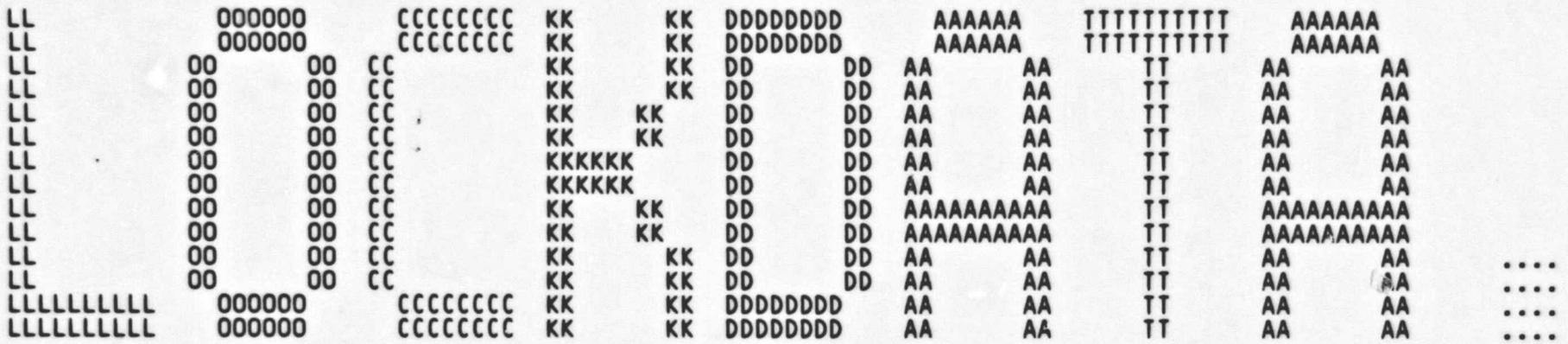

BBBBBBBBBBB OOOOOOOOO OOOOOOOOO TTTTTTTTTTTTTTT SSSSSSSSSSS
BBB BBB OOO OOO OOO OOO TTT SSS
BBBBBBBBBBB OOO OOO OOO OOO TTT SSSSSSSSS
BBB BBB OOO OOO OOO OOO TTT SSS
BBBBBBBBBBB OOOOOOOOO OOOOOOOOO TTT SSSSSSSSSSS

FILEID**LOCKDATA



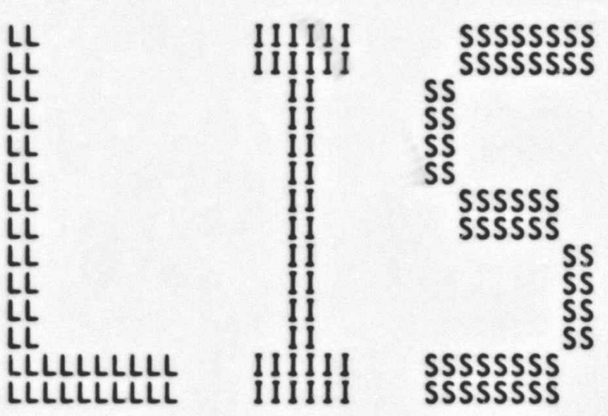


Table of contents

```
0000     1          .IF     NDF,STASW
0000     2          .TITLE  LOCKDATA - Routines to lock/unlock SYSGEN database
0000     3          .IFF
0000     4          .TITLE  STALOCK  - Dummy routines for STASYSGEN
0000     5          .ENDC
0000     6          .IDENT  'V04-000'
0000     7  ;
0000     8  ;*****************************************************************************
0000     9  ;*                                                                           *
0000    10  ;*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
0000    11  ;*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
0000    12  ;*  ALL RIGHTS RESERVED.                                                     *
0000    13  ;*                                                                           *
0000    14  ;*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
0000    15  ;*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
0000    16  ;*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
0000    17  ;*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
0000    18  ;*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
0000    19  ;*  TRANSFERRED.                                                             *
0000    20  ;*                                                                           *
0000    21  ;*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
0000    22  ;*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT    *
0000    23  ;*  CORPORATION.                                                             *
0000    24  ;*                                                                           *
0000    25  ;*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
0000    26  ;*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                  *
0000    27  ;*                                                                           *
0000    28  ;*                                                                           *
0000    29  ;*****************************************************************************
0000    30
0000    31  ;++
0000    32  ;
0000    33  ; Facility:  System generation and initialization
0000    34  ;
0000    35  ; Abstract:  LOCKDATA provides subroutines which allow synchronization of
0000    36  ;            access to the SYSGEN device database.
0000    37  ;
0000    38  ; Environment:
0000    39  ;
0000    40  ; Author:  Maryann Hinden, Creation date: 08-June-1983
0000    41  ;
0000    42  ; Modification History:
0000    43  ;
0000    44  ;       V03-002 WHM0001         Bill Matthews           29-Feb-1984
0000    45  ;               Enable queueing of the SYSGEN database lock request.
0000    46  ;
0000    47  ;       V03-001 MSH0001         Maryann Hinden          13-Jul-1983
0000    48  ;               Use general addressing mode for EXE$GL_SYSID_LOCK.
0000    49  ;--
0000    50
0000    51  ;
0000    52  ; Include files:
0000    53  ;
0000    54          $LCKDEF                         ; Define lock manager symbols
0000    55          $SSDEF                          ; Define system status values
0000    56          $SYSGMSGDEF                     ; Sysgen messages
0000    57
```

```
                                                B 1
                                    0000     58 ;
                                    0000     59 ; Equated Symbols:
                                    0000     60 ;
                           00000002 0000     61          LOCK_EFN = 2                    ; Event flag number used for lock request
                           00000018 0000     62          LOCK_FLAGS = LCK$M_SYNCSTS!-    ; Flags specified for lock request
                                    0000     63                       LCK$M_SYSTEM
                                    0000     64
                                00000000     65          .PSECT PAGED_DATA       NOEXE
                                    0000     66 ;
                                    0000     67 ;  Data for SYSGEN database lock
                                    0000     68 ;
                                    0000     69 BOO$GB_RESDSC::                          ; Descriptor for resource name
4E 45 47 53 59 53 00000008'010E0000' 0000     70          .ASCID  /SYSGEN$_DATABASE/
      45 53 41 42 41 54 41 44 5F 24  000E
                                    0018     71
                                    0018     72 BOO$LOCK_STATUS::                        ; LOCK_STATUS and LOCK_ID form
                           00000000 0018     73          .LONG   0                       ; the lock status block for the
                                    001C     74 BOO$LOCK_ID::                            ; lock request
                           00000000 001C     75          .LONG   0
                                    0020     76
                                00000000     77          .PSECT PAGED_CODE       NOWRT
```

```
                0000    79
```

```
                                        .SBTTL  BOO$LOCK_GEN - Lock SYSGEN database
                                ;++
                                ;  Functional description
                                ;       Lock the SYSGEN database for create/modify, in order to
                                ;       synchronize I/O database building.  Used by LOAD, RELOAD,
                                ;       AUTOCONFIGURE, CONNECT.
                                ;
                                ;       This routine attempts to acquire an exclusive mode system lock on
                                ;       the SYSGEN$_DATABASE resource in executive mode.  So that this resource
                                ;       is only specific to the local system, rather than a cluster, the $ENQ
                                ;       request specifies a parent (resource) which identifies the system on
                                ;       which the request is being made.
                                ;
                                ;       The $ENQW entry point is used, so that if the resource is not
                                ;       available immediately, the process will wait, with the assumption being
                                ;       that it will get the resource soon.  An event flag number is specified
                                ;       to avoid possible interference with CONFIGURE.
                                ;
                                ;       An alternate version of this routine (STALOCK) is provided in
                                ;       which the calls to the entry points simply return a success status.
                                ;       This is provided for use by STASYSGEN which runs standalone, and
                                ;       therefore no locking is necessary.
                                ;
                                ;  Calling sequence
                                ;       JSB/BSBx   BOO$LOCK_GEN
                                ;
                                ;  Inputs
                                ;       NONE
                                ;
                                ;  Outputs
                                ;       R0 - If LBS, locked database
                                ;            LBC, then no current access to database (SYSG$_NOLOCK)
                                ;--
                                
                                BOO$LOCK_GEN::
                                
                                        .IF     NDF, STASW
                                        $CMEXEC_S  LOCK                                 ; Change mode to take out lock
                           05   000F             RSB                                     ; Return
                                
                                LOCK:   .WORD   ^M<R2>                                  ; Null entry mask
 50    00000000'GF      D0     0012             MOVL    G^EXE$GL_SYSID_LOCK,R0          ; Get ID of parent lock
                                        $ENQW_S efn     = #LOCK_EFN,-
                                                lkmode  = #LCK$K_EXMODE,-
                                                lksb    = BOO$LOCK_STATUS,-
                                                flags   = #LOCK_FLAGS,-
                                                resnam  = BOO$GB_RESDSC,-
                                                parid   = R0
           0B 50      E9     003A             BLBC    R0,10$                          ; If LBC, error
 50    00000018'EF      3C     003D             MOVZWL  BOO$LOCK_STATUS,R0              ; Get final status
           01 50      E9     0044             BLBC    R0,10$                          ; Success?
                      04     0047             RET                                     ; Yes
 50    007C812A 8F      D0     0048     10$:    MOVL    #SYSG$_NOLOCK,R0                ; Indicate error
                      04     004F             RET
                             0050
                             0050             .IFF
                             0050
```

```
0050   138          MOVZWL  #SS$_NORMAL, R0                  ; Force success
0050   139          RSB
0050   140
0050   141          .ENDC
0050   142
```

```
                     0050   144          .SBTTL  BOO$UNLOCK_GEN - Unlock SYSGEN database
                     0050   145 ;++
                     0050   146 ;  Functional description
                     0050   147 ;       Dequeue the lock requested by BOO$LOCK_GEN.
                     0050   148 ;
                     0050   149 ;  Calling sequence
                     0050   150 ;       JSB/BSBx   BOO$UNLOCK_GEN
                     0050   151 ;
                     0050   152 ;  Input
                     0050   153 ;       Lock id in lock status block (implicit).
                     0050   154 ;
                     0050   155 ;  Output
                     0050   156 ;       R0 - If LBS, successful completion
                     0050   157 ;            LBC, error on dequeue (probably serious) - status SYSG$_DEQERR.
                     0050   158 ;--
                     0050   159
                     0050   160 BOO$UNLOCK_GEN::
                     0050   161
                     0050   162          .IF     NDF,STASW
                     0050   163
                     0050   164          $CMEXEC_S  UNLOCK                   ; Change mode to access lock
                  05 005F   165          RSB                                 ; Return
                     0060   166
                0000 0060   167 UNLOCK: .WORD   0                           ; Null entry mask
                     0062   168          $DEQ_S  lkid = BOO$LOCK_ID          ; Dequeue lock
              07 50   E8    0073   169          BLBS    R0,10$              ; If LBS, all okay
50   007C8122 8F   D0    0076   170          MOVL    #SYSG$_DEQERR,R0        ; Indicate error
                  04 007D   171 10$:     RET
                     007E   172
                     007E   173          .IFF
                     007E   174
                     007E   175          MOVZWL  #SS$_NORMAL,R0              ; Force success
                     007E   176          RSB
                     007E   177
                     007E   178          .ENDC
                     007E   179          .END
```

```
$$T1                            = 00000001
BOO$GB_RESDSC                     00000000 RG    02
BOO$LOCK_GEN                      00000000 RG    03
BOO$LOCK_ID                       0000001C RG    02
BOO$LOCK_STATUS                   00000018 RG    02
BOO$UNLOCK_GEN                    00000050 RG    03
EXE$GL_SYSID_LOCK                 ******** X     03
LCK$K_EXMODE                    = 00000005
LCK$M_SYNCSTS                   = 00000008
LCK$M_SYSTEM                    = 00000010
LOCK                              00000010 R     03
LOCK_EFN                        = 00000002
LOCK_FLAGS                      = 00000018
SYS$CMEXEC                        ******** GX    03
SYS$DEQ                           ******** GX    03
SYS$ENQW                          ******** GX    03
SYSG$_DEQERR                    = 007C8122
SYSG$_NOLOCK                    = 007C812A
UNLOCK                            00000060 R     03
```

+----------------+
! 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
PAGED_DATA	00000020	(32.)	02	(2.)	NOPIC	USR	CON	REL	LCL	NOSHR	NOEXE	RD	WRT	NOVEC BYTE
PAGED_CODE	0000007E	(126.)	03	(3.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC BYTE

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

Phase	Page faults	CPU Time	Elapsed Time
Initialization	36	00:00:00.08	00:00:00.37
Command processing	138	00:00:00.63	00:00:04.01
Pass 1	225	00:00:05.30	00:00:10.92
Symbol table sort	0	00:00:00.77	00:00:01.17
Pass 2	48	00:00:00.94	00:00:01.78
Symbol table output	4	00:00:00.04	00:00:00.04
Psect synopsis output	1	00:00:00.02	00:00:00.13
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	454	00:00:07.79	00:00:18.43

```
The working set limit was 1200 pages.
26796 bytes (53 pages) of virtual memory were used to buffer the intermediate code.
There were 30 pages of symbol table space allocated to hold 514 non-local and 2 local symbols.
179 source lines were read in Pass 1, producing 15 object records in Pass 2.
17 pages of virtual memory were used to define 16 macros.
```

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

Macro library name                                  Macros defined
------------------                                  --------------
_$255$DUA28:[BOOTS.OBJ]BOOTS.MLB;1                           0
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                               1
_$255$DUA28:[SYSLIB]STARLET.MLB;2                           12
TOTALS (all libraries)                                      13

649 GETS were required to define 13 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:LOCKDATA/OBJ=OBJ$:LOCKDATA MSRC$:LOCKDATA/UPDATE=(ENH$:LOCKDATA)+EXECML$/LIB+LIB$:BOOTS.MLB/LIB
```

CVBTDRIVR
LIS

DDBTDRIVR
LIS

DMBTDRIVR
LIS

DQBTDRIVR
LIS

CONIO
LIS

LOADERSUB
LIS

CONFIGUTL
LIS

DLBTDRIVR
LIS

DXBTDRIVR
LIS

INITPGFIL
LIS

LOCKDATA
LIS

LOADDRIV
LIS

LOADER
LIS

READLBN
LIS

RTFILREAD
LIS

SHARE
LIS

QUSS
LIS

RWBTDRIVR
LIS

MBBTDRIVR
LIS

SCSLOADER
LIS

RMSCONIO
LIS

READDRIV
LIS

PABTDRIVR
LIS

PUBTDRIVR
LIS

PUTERROR
LIS

READPRMPT
LIS