F 5

```
   SSSSSSSSSSSS   DDDDDDDDDDDD        AAAAAAAAA
   SSSSSSSSSSSS   DDDDDDDDDDDD        AAAAAAAAA
   SSSSSSSSSSSS   DDDDDDDDDDDD        AAAAAAAAA
SSS               DDD         DDD  AAA         AAA
SSS               DDD         DDD  AAA         AAA
SSS               DDD         DDD  AAA         AAA
SSS               DDD         DDD  AAA         AAA
SSS               DDD         DDD  AAA         AAA
SSS               DDD         DDD  AAA         AAA
   SSSSSSSSS      DDD         DDD  AAA         AAA
   SSSSSSSSS      DDD         DDD  AAA         AAA
   SSSSSSSSS      DDD         DDD  AAA         AAA
            SSS   DDD         DDD  AAAAAAAAAAAAAAA
            SSS   DDD         DDD  AAAAAAAAAAAAAAA
            SSS   DDD         DDD  AAAAAAAAAAAAAAA
            SSS   DDD         DDD  AAA         AAA
            SSS   DDD         DDD  AAA         AAA
            SSS   DDD         DDD  AAA         AAA
SSSSSSSSSSSS      DDDDDDDDDDDD     AAA         AAA
SSSSSSSSSSSS      DDDDDDDDDDDD     AAA         AAA
SSSSSSSSSSSS      DDDDDDDDDDDD     AAA         AAA
```

```
_S

Sy
--
CT
CT
CT
CT
CU
CU
CU
DC
DC
DC
DC
DC
DC
DE
DE
DE
DE
DE
DE
DE
DE
DI
DI
DI
DI
DI
DI
DI
DI
DI
DI
DI
DI
DI
DU
DU
DU
DU
DU
DU
DU
DU
DV
DV
DV
DY
DY
DY
DY
DY
ER
ES
```

```
**FILE**ID**LOCK                                                    M  4




LL            OOOOOO      CCCCCCCC  KK      KK
LL            OOOOOO      CCCCCCCC  KK      KK
LL          OO      OO  CC          KK      KK
LL          OO      OO  CC          KK      KK
LL          OO      OO  CC          KK    KK
LL          OO      OO  CC          KK    KK
LL          OO      OO  CC          KKKKKK
LL          OO      OO  CC          KKKKKK
LL          OO      OO  CC          KK    KK
LL          OO      OO  CC          KK    KK
LL          OO      OO  CC          KK      KK       ....
LL          OO      OO  CC          KK      KK       ....
LLLLLLLLLL    OOOOOO      CCCCCCCC  KK      KK       ....
LLLLLLLLLL    OOOOOO      CCCCCCCC  KK      KK       ....

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
```

```
LOCK
V04-

65 6

   6

65 6
```

Table of contents

```
0000          .TITLE  LOCK LOCK AND RESOURCE FORMATTING ROUTINES
0000          .SBTTL  COPYRIGHT NOTICE
0000          .IDENT  'V04-000'
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        .SBTTL  PROGRAM DESCRIPTION
0000  ;++
0000  ; FACILITY
0000  ;
0000  ;     SYSTEM DUMP ANALYZER
0000  ;
0000  ; ABSTRACT
0000  ;
0000  ;     ROUTINES TO FORMAT LOCK AND RESOURCE INFORMATION
0000  ;
0000  ; ENVIRONMENT
0000  ;
0000  ;     NATIVE MODE, USER MODE
0000  ;
0000  ; AUTHOR
0000  ;
0000  ;     MARYANN HINDEN, JUNE 1982
0000  ;
0000  ; MODIFIED BY
0000  ;
0000  ;     V03-007 SRB0109         Steve Beckhardt         2-Feb-1984
0000  ;             Fixed formatting of Directory entry text in SHO RES line.
0000  ;
0000  ;     V03-006 SRB0107         Steve Beckhardt         7-Dec-1983
0000  ;             Enhanced SHOW LOCK and SHOW RESOURCE to display
0000  ;             distributed lock information.
0000  ;
0000  ;     V03-005 SRB0089         Steve Beckhardt         27-May-1983
0000  ;             Fixed bug displaying resource name.  Added new status bit
0000  ;             definitions.
0000  ;
0000  ;     V03-004 SRB0071         Steve Beckhardt         16-Mar-1983
0000  ;             Changed the way system resources are decoded in RSB
0000  ;             to use zero group number instead of SYSNAM flag.
0000  ;
0000  ;     V03-003 MSH0003         Maryann Hinden          17-Dec-1982
0000  ;             Fix broken assume's.
0000  ;
0000  ;     V03-002 MSH0002         Maryann Hinden          22-Oct-1982
0000  ;             Determine empty lock queue correctly in SHOW_PROC_LOCK.
0000  ;             Change lock display - output lock and parent id in hex;
0000  ;             rearrange format.
0000  ;
0000  ;     V03-001 MSH0001         Maryann Hinden          16-Jul-1982
0000  ;             Correct references to LCK$xxxx data structures; change
0000  ;             condition handling.
0000  ;--
```

```
                  0000     77            .SBTTL  DECLARATIONS
                  0000     78 ;
                  0000     79 ;          SYSTEM SYMBOL DEFINTIONS
                  0000     80 ;
                  0000     81
                  0000     82            $CHFDEF
                  0000     83            $LCKDEF
                  0000     84            $LKBDEF
                  0000     85            $OPTDEF
                  0000     86            $PCBDEF
                  0000     87            $RSBDEF
                  0000     88            $STSDEF
                  0000     89
                  0000     90 ;
                  0000     91 ;          LOCAL SYMBOL DEFINITIONS
                  0000     92 ;
                  0000     93
        FFFFFFFF  0000     94 GRANT    = -1              ; SYMBOLIC DESIGNATIONS FOR QUEUE TYPES
        00000000  0000     95 CONVERT  = 0
        00000001  0000     96 WAIT     = 1
                  0000     97
        00000000  0000     98 LKID_OFF = LKB$L_LKID-LKB$L_LKID         ; OFFSETS INTO SAVED QUEUE DATA
        00000005  0000     99 GRMD_OFF = LKB$B_GRMODE-LKB$L_LKID
        00000004  0000    100 RQMD_OFF = LKB$B_RQMODE-LKB$L_LKID
                  0000    101
```

```
                                    0000   103            .SBTTL  STORAGE DEFINITIONS
                                00000000   104            .PSECT  SDADATA,NOEXE,WRT,LONG
                                    0000   105 ;
                                    0000   106 ;          Data area for SHOW LOCKS commands
                                    0000   107 ;
                                    0000   108
                                    0000   109 LOCK_COUNT:                                  ; count of locks processed
                          00000000  0000   110            .LONG   0
                                    0004   111 PARID_BFR:                                   ; parent lock id
                          00000000  0004   112            .LONG   0
                                    0008   113 LKB_BFR:                                     ; lock block data
                          00000068  0008   114            .BLKB   LKB$K_LENGTH
                                    0068   115
                                    0068   116 RSBCSID_BFR:                                 ; CSID in RSB
                          0000006C  0068   117            .BLKL   1
                                    006C   118 LKB_RSB_BFR:                                 ; resource block data for given lock
                                    006C   119 GROUP_BFR:                                   ;       group number
                          0000006E  006C   120            .BLKW   1
                                    006E   121 RMOD_BFR:                                    ;       access mode
                          0000006F  006E   122            .BLKB   1
                                    006F   123 RSNLEN_BFR:                                  ;       resource name length
                          00000070  006F   124            .BLKB   1
                                    0070   125 RESNAM_BFR:                                  ;       resource name
                          00000090  0070   126            .BLKB   RSB$K_MAXLEN+1
                                    0090   127
                                    0090   128 LOCKID::                                     ; lock ID currently being processed
                          00000094  0090   129            .BLKL   1
                                    0094   130
                                    0094   131            .ALIGN  LONG
                                    0094   132
                                    0094   133 FAO_RMINFO_DSC:
63 6F 6C 20 66 6F 0000009C'010E0000'  0094   134            .ASCID  "of lock !XL on system !XL"
73 79 73 20 6E 6F 20 4C 58 21 20 6B  00A2
                  4C 58 21 20 6D 65 74  00AE
                                    00B5   135
                                    00B5   136            .ALIGN  LONG
                                    00B8   137 RMINFO_LEN:
                          00000000  00B8   138            .LONG   0
                                    00BC   139 RMINFO_DSC:
                          00000050  00BC   140            .LONG   80
                          000000C5' 00C0   141            .ADDRESS        RMINFO_BFR
                                    00C4   142
                                    00C4   143 RMINFO_CNT:
                          000000C5  00C4   144            .BLKB   1
                                    00C5   145 RMINFO_BFR:
                          00000115  00C5   146            .BLKB   80
                                    0115   147
                                    0115   148 ;
                                    0115   149 ;          Data area for SHOW RESOURCES commands
                                    0115   150 ;
                                    0115   151
                                    0115   152 RES_COUNT:                                   ; count of resources processed
                          00000000  0115   153            .LONG   0
                                    0119   154 QUEUE_COUNT:                                 ; count of queue elements processed
                          00000000  0119   155            .LONG   0
                                    011D   156 RSB_BFR:                                     ; resource block data
                          0000018C  011D   157            .BLKB   RSB$K_MAXLEN+RSB$K_LENGTH
```

```
                                 018C  HTBL_INDX:                                   ; current index into hash table
                       00000190  018C             .BLKL   1
                                 0190  HTBL_CNT:                                    ; size of hash table
                       00000194  0190             .BLKL   1
                                 0194  COUNT:                                       ; # queue items to display
                       00000198  0194             .BLKL   1
                                 0198  QUEUE_TYPE:                                  ; type of queue being processed
                       0000019C  0198             .BLKL   1
                                 019C  GRMD_BFR:                                    ; grant mode
                       0000019F  019C             .BLKB   3
                                 019F  RQMD_BFR:                                    ; requested mode
                       000001A2  019F             .BLKB   3
                                 01A2  LKID_BFR:                                    ; lock id
                       000001AE  01A2             .BLKL   3
                                 01AE  FAO_GROUP_DSC:
     57 4F 33 21 000001B6'010E0000'  01AE             .ASCID  "!3OW"
                                 01BA
                                 01BA  GROUP_BUF:
                       000001BD  01BA             .BLKB   3
                                 01BD
                                 01BD             .ALIGN  LONG
                                 01C0  GROUP_BUF_DSC:
                       00000003  01C0             .LONG   3
                       000001BA' 01C4             .ADDRESS        GROUP_BUF
                                 01C8
                                 01C8  SYS_DSC:
6D 65 74 73 79 53 000001D0'010E0000'  01C8             .ASCID  "System"
                                 01D6
                                 01D6             .ALIGN  LONG
                                 01D8  GROUP_DSC:
                       0000000B  01D8             .LONG   11
                       000001E0' 01DC             .ADDRESS        GROUP_TXT
                                 01E0
                                 01E0  GROUP_TXT:
         20 20 20 70 75 6F 72 47  01E0             .ASCII  "Group   "
                                 01E8  GROUP_NUM:
                       000001EB  01E8             .BLKB   3
                                 01EB
                                 01EB  ;
                                 01EB  ;          FAO data storage for RESOURCE display
                                 01EB  ;
                                 01EB
                                 01EB  ;  line 1
                                 01EB
                       00000000  01EB  RSB:      .LONG   0               ; Address of RSB
                       00000000  01EF  GGMOD:    .LONG   0               ; Group grant mode (addr of ASCIC string)
                                 01F3
                                 01F3  ;  line 2
                                 01F3
                       00000000  01F3  PRSB:     .LONG   0               ; Address of parent RSB
                       00000000  01F7  CGMOD:    .LONG   0               ; Conversion grant mode (addr of ASCIC string)
                                 01FB
                                 01FB  ;  line 3
                                 01FB
                       00000000  01FB  SRSBCT:   .LONG   0               ; Sub-RSB reference count
                       00000000  01FF  BAST:     .LONG   0               ; Blocking AST count
                                 0203
```

```
                    0203   215 ; line 4
                    0203   216
          00000000  0203   217 VAL4:   .LONG   0                   ; Value block longword #4
          00000000  0207   218 VAL3:   .LONG   0                   ; Value block longword #3
          00000000  020B   219 VAL2:   .LONG   0                   ; Value block longword #2
          00000000  020F   220 VAL1:   .LONG   0                   ; Value block longword #1
          00000000  0213   221 SEQNUM: .LONG   0                   ; Sequence number
          00000000  0217   222 VALID:  .LONG   0                   ; Value block valid (addr. of .ASCIC string)
                    021B   223
                    021B   224 ; line 5
                    021B   225
          00000000  021B   226 RESN2:  .LONG   0                   ; Second longword of resource name
          00000000  021F   227 RESN1:  .LONG   0                   ; First    "       "     "       "
          00000008  0223   228         .LONG   8                   ; Count of ASCII string
          00000000  0227   229 BUF1:   .LONG   0                   ; Text of resource name (addr of ASCII string)
                    022B   230
                    022B   231 ; line 6
                    022B   232
          00000000  022B   233 RESNLEN:.LONG   0                   ; Resource name length
          00000000  022F   234 RESN4:  .LONG   0                   ; Fourth longword of resource name
          00000000  0233   235 RESN3:  .LONG   0                   ; Third    "       "     "       "
          00000008  0237   236         .LONG   8                   ; Count of ASCII string
          00000000  023B   237 BUF2:   .LONG   0                   ; Text of resource name (addr of ASCII string)
          00000000  023F   238 CSID:   .LONG   0                   ; RSB CSID
                    0243   239
                    0243   240 ; line 7
                    0243   241
          00000000  0243   242 RACMOD: .LONG   0                   ; Resource access mode (addr of ASCIC string)
          00000000  0247   243 RESN6:  .LONG   0                   ; Sixth longword of resource name
          00000000  024B   244 RESN5:  .LONG   0                   ; Fifth    "       "     "       "
          00000008  024F   245         .LONG   8                   ; Count of ASCII string
          00000000  0253   246 BUF3:   .LONG   0                   ; Text of resource name (addr of ASCII string)
                    0257   247 DIRENTRY:
          00000000  0257   248         .LONG   0                   ; Directory entry (addr. of .ASCIC string)
                    025B   249
                    025B   250 ; line 8
                    025B   251
          00000000  025B   252 RNSPACE: .LONG  0                   ; Address of descriptor
          00000000  025F   253 RESN8:  .LONG   0                   ; Eighth longword of resource name
          00000000  0263   254 RESN7:  .LONG   0                   ; Seventh  "       "     "       "
          00000008  0267   255         .LONG   8                   ; Count of ASCII string
          00000000  026B   256 BUF4:   .LONG   0                   ; Text of resource name (addr of ASCII string)
                    026F   257
```

```
           026F  ;
           026F  ;        FAO data storage for LOCKS display
           026F  ;
           026F
           026F  ;  LINE 1
           026F
00000000   026F  LKID:    .LONG   0               ; Lock id (value)
00000000   0273  PID:     .LONG   0               ; PID (value)
00000000   0277  FLAGS1:  .LONG   0               ; First line of flags (addr. of .ASCIC string)
00000000   027B           .LONG   0               ;   "     "   "   "
00000000   027F           .LONG   0               ;   "     "   "   "
           0283
           0283  ;  LINE 2
           0283
00000000   0283  PARID:   .LONG   0               ; Parent id (value)
00000000   0287  STATE1:  .LONG   0               ; Lock state info (address of .ASCIC string)
00000002   028B           .LONG   2               ; Lock mode (length of string)
00000000   028F  MODE1:   .LONG   0               ; Lock mode (address of .ASCII string)
00000000   0293  FLAGS2:  .LONG   0               ; Second line of flags (addr. of .ASCIC string)
00000000   0297           .LONG   0               ;   "     "   "   "
00000000   029B           .LONG   0               ;   "     "   "   "
           029F
           029F  ;  LINE 3
           029F
00000000   029F  SUBLKS:  .LONG   0               ; Sublocks (value)
00000000   02A3  STATE2:  .LONG   0               ; More lock state info (addr. of .ASCIC string)
00000002   02A7           .LONG   2               ; More lock mode (length of string)
00000000   02AB  MODE2:   .LONG   0               ; More lock mode ( addr. of .ASCII string)
00000000   02AF  FLAGS3:  .LONG   0               ; Third line of flags (addr. of .ASCIC string)
00000000   02B3           .LONG   0               ;   "     "   "   "
00000000   02B7           .LONG   0               ;   "     "   "   "
           02BB
           02BB  ;  LINE 3A
           02BB
00000000   02BB  BLKAST:  .LONG   0               ; Blocking AST (address of .ASCIC string)
00000000   02BF  LKB:     .LONG   0               ; Address of LKB
           02C3
           02C3  ;  LINE 4
           02C3
00000000   02C3  RES2:    .LONG   0               ; Second longword of resource name (value)
00000000   02C7  RES1:    .LONG   0               ; First longword of resource name (value)
00000008   02CB           .LONG   8
00000000   02CF  DMP1:    .LONG   0               ; First line of text name (.ASCII string)
00000000   02D3  STS1:    .LONG   0               ; 1st line of status (addr. of .ASCIC strings)
00000000   02D7           .LONG   0
00000000   02DB           .LONG   0
           02DF
           02DF  ;  LINE 5
           02DF
00000000   02DF  RLEN:    .LONG   0               ; Resource name length (value)
00000000   02E3  RES4:    .LONG   0               ; Fourth longword of resource name (value)
00000000   02E7  RES3:    .LONG   0               ; Third longword of resource name (value)
00000008   02EB           .LONG   8
00000000   02EF  DMP3:    .LONG   0               ; Second line of text name (.ASCII string)
00000000   02F3  STS2:    .LONG   0               ; 2nd line of status (addr. of .ASCIC strings)
00000000   02F7           .LONG   0
00000000   02FB           .LONG   0
```

```
          02FF
          02FF  ; LINE 6
          02FF
00000000  02FF  ACMODE: .LONG   0                    ; Access mode (Address of .ASCIC string)
00000000  0303  RES6:   .LONG   0                    ; Sixth longword of resource name (value)
00000000  0307  RES5:   .LONG   0                    ; Fifth longword of resource name (value)
00000008  030B          .LONG   8
00000000  030F  DMP5:   .LONG   0                    ; Third line of text name (.ASCII string)
00000000  0313  STS3:   .LONG   0                    ; 3rd line of status (addr. of .ASCIC strings)
00000000  0317          .LONG   0
00000000  031B          .LONG   0
          031F
          031F  ; LINE 7
          031F
00000000  031F  NSPACE: .LONG   0                    ; Name space (group/system) (addr. of desc.)
00000000  0323  RES8:   .LONG   0                    ; Eighth longword of resource name (value)
00000000  0327  RES7:   .LONG   0                    ; Seventh longword of resource name (value)
00000008  032B          .LONG   8
00000000  032F  DMP7:   .LONG   0                    ; Fourth line of text name (.ASCII string)
          0333
          0333  ; LINE 8
          0333
00000000  0333  TYPE:   .LONG   0                    ; Lock type (addr. of .ASCIC string)
00000000  0337  RMINFO: .LONG   0                    ; Remote info (addr. of .ASCIC string)
          033B
```

```
                                                033B   342             .SBTTL  READ-ONLY DATA DEFINITIONS
                                            00000000   343             .PSECT  LOCK,EXE,NOWRT,LONG
                                                0000   344
                                                0000   345 ;  HEADERS
                                                0000   346
                                                0000   347 LOCK_HEAD:  STRING <Lock database>
                                                0015   348 RES_HEAD:   STRING <Resource database>
                                                002E   349
                                                002E   350 LOCK_STR_TBL:                        ; table of FAO strings for wait/grant queue display
                                   00000000     002E   351             .LONG   0
                                   0000005A'    0032   352             .LONG   LOCKSTR1
                                   0000006F'    0036   353             .LONG   LOCKSTR2
                                   00000093'    003A   354             .LONG   LOCKSTR3
                                                003E   355
                                                003E   356 CONV_STR_TBL:                        ; table of FAO strings for convert queue display
                                   00000000     003E   357             .LONG   0
                                   000000C6'    0042   358             .LONG   CONVSTR1
                                   000000DF'    0046   359             .LONG   CONVSTR2
                                   00000108'    004A   360             .LONG   CONVSTR3
                                                004E   361
                                                004E   362 QUE_STR_TBL:                         ; table of headers for queue display
                                   00000141'    004E   363             .LONG   GRANTSTR
                                   0000016B'    0052   364             .LONG   CONVSTR
                                   0000019B'    0056   365             .LONG   WAITSTR
                                                005A   366
21 20 20 20 20 20 00000062'010E0000'   005A   367 LOCKSTR1:   .ASCID      #     !XL  !AC#
                  43 41 21 20 20 4C 58   0068
21 20 20 20 20 20 00000077'010E0000'   006F   368 LOCKSTR2:   .ASCID      #     !XL  !AC       !XL  !AC#
20 20 20 20 20 43 41 21 20 20 4C 58   007D
      43 41 21 20 20 4C 58 21 20 20   0089
21 20 20 20 20 20 0000009B'010E0000'   0093   369 LOCKSTR3:   .ASCID      #     !XL  !AC       !XL  !AC       !XL  !AC#
20 20 20 20 20 43 41 21 20 20 4C 58   00A1
20 20 43 41 21 20 20 4C 58 21 20 20   00AD
41 21 20 20 4C 58 21 20 20 20 20 20   00B9
                                  43   00C5
                                       00C6   370
21 20 20 20 20 20 000000CE'010E0000'   00C6   371 CONVSTR1:   .ASCID      #     !XL  !AC/!AC#
   43 41 21 2F 43 41 21 20 20 4C 58   00D4
21 20 20 20 20 20 000000E7'010E0000'   00DF   372 CONVSTR2:   .ASCID      #     !XL  !AC/!AC    !XL  !AC/!AC#
20 43 41 21 2F 43 41 21 20 20 4C 58   00ED
2F 43 41 21 20 20 4C 58 21 20 20 20   00F9
                            43 41 21   0105
21 20 20 20 20 20 00000110'010E0000'   0108   373 CONVSTR3:   .ASCID      #     !XL  !AC/!AC    !XL  !AC/!AC    !XL  !AC/!AC#
20 43 41 21 2F 43 41 21 20 20 4C 58   0116
2F 43 41 21 20 20 4C 58 21 20 20 20   0122
20 20 4C 58 21 20 20 20 20 43 41 21   012E
                  43 41 21 2F 43 41 21   013A
                                       0141   374
65 74 6E 61 72 47 00000149'010E0000'   0141   375 GRANTSTR:   .ASCID      #Granted queue (Lock ID / Gr mode):#
63 6F 4C 28 20 65 75 65 75 71 20 64   014F
6F 6D 20 72 47 20 2F 20 44 49 20 6B   015B
                        3A 29 65 64   0167
72 65 76 6E 6F 43 00000173'010E0000'   016B   376 CONVSTR:    .ASCID      #Conversion queue (Lock ID / Gr/Rq mode):#
28 20 65 75 65 75 71 20 6E 6F 69 73   0179
72 47 20 2F 20 44 49 20 6B 63 6F 4C   0185
      3A 29 65 64 6F 6D 20 71 52 2F   0191
6E 69 74 69 61 57 000001A3'010E0000'   019B   377 WAITSTR:    .ASCID      #Waiting queue (Lock ID / Rq mode):#
```

```
63 6F 4C 28 20 65 75 65 75 71 20 67  01A9
6F 6D 20 71 52 20 2F 20 44 49 20 6B  01B5
                        3A 29 65 64  01C1
                                     01C5    378
                                     01C5    379 ;
                                     01C5    380 ;         Generic Data
                                     01C5    381 ;
                                     01C5    382 LOCKMODE_TBL:
                              4C 4E  01C5    383           .ASCII  "NL"
                              52 43  01C7    384           .ASCII  "CR"
                              57 43  01C9    385           .ASCII  "CW"
                              52 50  01CB    386           .ASCII  "PR"
                              57 50  01CD    387           .ASCII  "PW"
                              58 45  01CF    388           .ASCII  "EX"
                                     01D1    389
                          000001E9'  01D1    390 LKMODE_TBL:      .ADDRESS NULL                                ; POINTERS TO MODE TEXT
                          000001EC'  01D5    391                  .ADDRESS CREAD
                          000001EF'  01D9    392                  .ADDRESS CWRITE
                          000001F2'  01DD    393                  .ADDRESS PREAD
                          000001F5'  01E1    394                  .ADDRESS PWRITE
                          000001F8'  01E5    395                  .ADDRESS EX
                          4C 4E 00'  01E9    396 NULL:    .ASCIC  /NL/
                                02   01E9
                          52 43 00'  01EC    397 CREAD:   .ASCIC  /CR/
                                02   01EC
                          57 43 00'  01EF    398 CWRITE:  .ASCIC  /CW/
                                02   01EF
                          52 50 00'  01F2    399 PREAD:   .ASCIC  /PR/
                                02   01F2
                          57 50 00'  01F5    400 PWRITE:  .ASCIC  /PW/
                                02   01F5
                          58 45 00'  01F8    401 EX:      .ASCIC  /EX/
                                02   01F8
                                     01FB    402
                                     01FB    403 GR_STRING:
   74 61 20 64 65 74 6E 61 72 47 00' 01FB    404          .ASCIC  "Granted at"
                                0A   01FB
                                     0206    405 CV_STRING:
20 67 6E 69 74 72 65 76 6E 6F 43 00' 0206    406          .ASCIC  "Converting to"
                             6F 74   0212
                                0D   0206
                                     0214    407 WT_STRING:
72 6F 66 20 67 6E 69 74 69 61 57 00' 0214    408          .ASCIC  "Waiting for"
                                0B   0214
                                     0220    409
                             20 20   0220    410 BLANKS: .ASCII  "  "
                                     0222    411
                                     0222    412 NULL_CSTRING:
                                00'  0222    413          .ASCIC  ""
                                00   0222
                                     0223    414
                                     0223    415 BL_STRING:
         54 53 41 4B 4C 42 00'       0223    416          .ASCIC  "BLKAST"
                                06   0223
                                     022A    417
                                     022A    418 FLAGS_TBL:
         20 4B 4C 42 4C 41 56 00'    022A    419          .ASCIC  "VALBLK "
```

```
                          07  022A
       54 52 45 56 4E 4F 43 00'  0232   420           .ASCIC  "CONVERT"
                          07  0232
       45 55 45 55 51 4F 4E 00'  023A   421           .ASCIC  "NOQUEUE"
                          07  023A
       53 54 53 43 4E 59 53 00'  0242   422           .ASCIC  "SYNCSTS"
                          07  0242
       20 4D 45 54 53 59 53 00'  024A   423           .ASCIC  "SYSTEM "
                          07  024A
       41 54 4F 55 51 4F 4E 00'  0252   424           .ASCIC  "NOQUOTA"
                          07  0252
       20 53 59 53 54 56 43 00'  025A   425           .ASCIC  "CVTSYS "
                          07  025A
       52 45 56 4F 43 45 52 00'  0262   426           .ASCIC  "RECOVER"
                          07  0262
       54 43 45 54 4F 52 50 00'  026A   427           .ASCIC  "PROTECT"
                          07  026A
                              0272   428
                              0272   429 STATUS_TBL:
       54 53 41 4C 50 43 44 00'  0272   430           .ASCIC  "DCPLAST"
                          07  0272
       54 53 41 4B 4C 42 44 00'  027A   431           .ASCIC  "DBLKAST"
                          07  027A
       20 20 43 4E 59 53 41 00'  0282   432           .ASCIC  "ASYNC  "
                          07  0282
       44 51 54 53 41 4C 42 00'  028A   433           .ASCIC  "BLASTQD"
                          07  028A
       20 59 50 43 54 53 4D 00'  0292   434           .ASCIC  "MSTCPY "
                          07  0292
       41 54 4F 55 51 4F 4E 00'  029A   435           .ASCIC  "NOQUOTA"
                          07  029A
       51 54 55 4F 4D 49 54 00'  02A2   436           .ASCIC  "TIMOUTQ"
                          07  02A2
       20 53 59 53 53 41 57 00'  02AA   437           .ASCIC  "WASSYS "
                          07  02AA
       53 59 53 4F 54 56 43 00'  02B2   438           .ASCIC  "CVTOSYS"
                          07  02B2
                              02BA   439
             6C 61 63 6F 4C 00'  02BA   440 LOCAL:    .ASCIC  "Local"
                          05  02BA
                              02C0   441 PROCESS:
       73 73 65 63 6F 72 50 00'  02C0   442           .ASCIC  "Process"
                          07  02C0
          72 65 74 73 61 4D 00'  02C8   443 MASTER:   .ASCIC  "Master"
                          06  02C8
                              02CF   444
                         20 00'  02CF   445 SPACE:    .ASCIC  " "
                          01  02CF
                              02D1   446
                              02D1   447 NOT_VALID:
    64 69 6C 61 76 20 74 6F 4E 00'  02D1   448           .ASCIC  "Not valid"
                          09  02D1
                              02DB   449 DIR_ENTRY:
65 20 79 72 6F 74 63 65 72 69 44 00'  02DB   450           .ASCIC  "Directory entry"
                    79 72 74 6E  02E7
                          0F  02DB
                              02EB   451
```

```
65 64 6F 6D 20 6C 65 6E 72 65 4B 00' 02EB     452 KMODE:   .ASCIC  "Kernel mode"
                                  0B  02EB
   65 64 6F 6D 20 2E 63 65 78 45 00' 02F7     453 EMODE:   .ASCIC  "Exec. mode"
                                  0A  02F7
65 64 6F 6D 20 2E 72 65 70 75 53 00' 0302     454 SMODE:   .ASCIC  "Super. mode"
                                  0B  0302
      65 64 6F 6D 20 72 65 73 55 00' 030E     455 UMODE:   .ASCIC  "User mode"
                                  09  030E
                                      0318    456
                                      0318    457          .ALIGN  LONG
                                      0318    458 ACMODE_TBL:
                           000002EB'  0318    459          .ADDRESS        KMODE
                           000002F7'  031C    460          .ADDRESS        EMODE
                           00000302'  0320    461          .ADDRESS        SMODE
                           0000030E'  0324    462          .ADDRESS        UMODE
                                      0328    463
```

```
                          0328          .SBTTL  SHOW_ALL_LOCKS - ACTION ROUTINE TO DISPLAY ALL LOCKS
                          0328    ;+++
                          0328    ;
                          0328    ;       SHOW_ALL_LOCKS
                          0328    ;
                          0328    ; PURPOSE
                          0328    ;       ACTION ROUTINE FOR THE "SHOW LOCKS/ALL" COMMAND
                          0328    ;
                          0328    ; INPUT
                          0328    ;       NONE
                          0328    ;
                          0328    ; OUTPUT
                          0328    ;       DISPLAYED DATA
                          0328    ;
                          0328    ;---
                    0000  0328          .ENTRY  SHOW_ALL_LOCKS,^M<>
                          032A
6D   000009E8'EF    9E    032A          MOVAB   LOCK_COND_HAND,(FP)     ; ESTABLISH CONDITION HANDLER
     00000090'EF    D4    0331          CLRL    LOCKID                  ; INITIALIZE VARIABLE
          FCC5 CF   9F    0337          PUSHAB  LOCK_HEAD               ; GET HEADER
00000000'EF    01   FB    033B          CALLS   #1,SET_HEADING          ; SET IT UP
     00000000'EF    D4    0342          CLRL    LOCK_COUNT              ; INIT COUNT OF LOCKS PROCESSED
                          0348
                          0348    ;
                          0348    ; STEP THROUGH LOCK ID TABLE, DISPLAYING ENTRIES WITH DATA
                          0348    ;
     00000090'EF    D6    0348    10$:  INCL    LOCKID                  ; INCREMENT INDEX
00000443'EF    00   FB    034E          CALLS   #0,GET_LKB              ; GET ADDRESS OF ASSOCIATED LKB
            F0 50   E9    0355          BLBC    R0,10$                  ; IF LBC, NON-EXISTENT, TRY NEXT
     00000000'EF    D6    0358          INCL    LOCK_COUNT              ; GOT ONE
000004B7'EF    00   FB    035E          CALLS   #0,SAVE_LOCK_DATA       ; GET ASSOCIATED DATA
0000051D'EF    00   FB    0365          CALLS   #0,DISPLAY_LOCK         ; DISPLAY DATA FOR THIS LOCKID
            DA      11    036C          BRB     10$                     ; LOOP (SIGNAL WILL EXIT FOR US)
```

```
                           036E          .SBTTL  SHOW_ONE_LOCK - ACTION ROUTINE TO DISPLAY ONE LOCK
                           036E    ;+++
                           036E    ;
                           036E    ;         SHOW_ONE_LOCK
                           036E    ;
                           036E    ;   PURPOSE
                           036E    ;         ACTION ROUTINE TO PROCESS "SHOW LOCK lockid" COMMAND
                           036E    ;
                           036E    ;   INPUT
                           036E    ;         LOCKID - INDEX INTO LOCK ID TABLE
                           036E    ;
                           036E    ;   OUTPUT
                           036E    ;         DISPLAYED DATA
                           036E    ;
                           036E    ;---
                     0000  036E          .ENTRY  SHOW_ONE_LOCK,^M<>
                           0370
00000443'EF   00     FB    0370          CALLS   #0,GET_LKB                ; GET LOCK BLOCK ADDRESS
         21   50     E9    0377          BLBC    R0,10$                    ; IF LBC, NO LOCK BLOCK FOR ID
                           037A
000004B7'EF   00     FB    037A          CALLS   #0,SAVE_LOCK_DATA         ; SAVE LOCK DATA
       FC7B   CF     9F    0381          PUSHAB  LOCK_HEAD                 ; SET UP LOCK HEADING
00000000'EF   01     FB    0385          CALLS   #1,SET_HEADING
0000051D'EF   00     FB    038C          CALLS   #0,DISPLAY_LOCK           ; DISPLAY LOCK INFO
                           0393          STATUS  SUCCESS                   ; SUCCESSFUL COMPLETION
                     04    039A          RET
                           039B
                           039B    10$:  SIGNAL  0,NOLKB                   ; NO LOCK BLOCK FOR THIS ID
                     04    03AD          RET
```

```
                                   03AE          .SBTTL  SHOW_PROC_LOCK - SHOW LOCKS ASSOCIATED WITH A PROCESS
                                   03AE  ;+++
                                   03AE  ;
                                   03AE  ;       SHOW_PROC_LOCK
                                   03AE  ;
                                   03AE  ; PURPOSE
                                   03AE  ;       SHOW LOCKS ASSOCIATED WITH A GIVEN LOCK.  CALLED FROM
                                   03AE  ;       GENERIC CODE WHICH DISPLAYS PROCESS DATA.
                                   03AE  ;
                                   03AE  ; INPUT
                                   03AE  ;       4(AP) - ADDRESS OF PCB COPY IN SDA IMAGE
                                   03AE  ;       8(AP) - ADDRESS OF "REAL" PCB IN SYSTEM
                                   03AE  ;
                                   03AE  ; OUTPUT
                                   03AE  ;       DISPLAYED DATA
                                   03AE  ;---
                             000C  03AE          .ENTRY  SHOW_PROC_LOCK,^M<R2,R3>
                                   03B0
               53     04 AC    D0  03B0          MOVL    4(AP),R3                   ; GET ADDRESS OF PCB SAVE AREA
            53      0104 C3    D0  03B4          MOVL    PCB$L_LOCKQFL(R3),R3       ; GET FORWARD LINK
               52     08 BC    D0  03B9          MOVL    @8(AP),R2                  ; GET "REAL" ADDRESS OF PCB
            52      0104 C2    DE  03BD          MOVAL   PCB$L_LOCKQFL(R2),R2       ; GET "REAL" ADDRESS OF HEAD OF QUEUE
                    52    53   D1  03C2          CMPL    R3,R2                      ; ARE THERE ELEMENTS IN THIS QUEUE?
                          69   13  03C5          BEQL    NOLCK                      ; IF EQL, NO
                                   03C7
000002BF'EF   53   00000040 8F C3  03C7  GETLCK: SUBL3   #LKB$L_OWNQFL,R3,LKB       ; ADJUST FOR START OF LKB
          000004B7'EF      00  FB  03D3          CALLS   #0,SAVE_LOCK_DATA          ; SAVE LOCK DATA
                                   03DA          SKIP    PAGE
                                   03E1          PRINT   0,<Lock data:>
          00000556'EF      00  FB  03EE          CALLS   #0,FORMAT_LOCK             ; FORMAT LOCK DATA
                                   03F5          ENSURE  9
                                   040D          SKIP    1
          00000856'EF      00  FB  0416          CALLS   #0,PRINT_LOCK              ; DISPLAY LOCK INFO
                                   041D          REQMEM  (R3),R3                    ; GET NEXT QUEUE LINK
                    52    53   D1  0429          CMPL    R3,R2                      ; AT END OF QUEUE?
                          14   13  042C          BEQL    DONE                       ; IF EQL, DONE
                          97   11  042E          BRB     GETLCK                     ; LOOP
                                   0430
                                   0430  NOLCK:                                     ; THIS PROCESS HAS NO LOCKS
                                   0430          SIGNAL  0,NOPRLOCK
                                   0442
                               04  0442  DONE:   RET                                ; SUCCESSFUL
```

```
                                0443          .SBTTL  GET_LKB - GET LOCK BLOCK
                                0443  ;+++
                                0443  ;
                                0443  ;       GET_LKB
                                0443  ;
                                0443  ;  PURPOSE
                                0443  ;       GIVEN LOCK ID, GET ADDRESS OF LOCK BLOCK.
                                0443  ;
                                0443  ;  INPUT
                                0443  ;       LOCKID - INDEX INTO LOCK ID TABLE
                                0443  ;
                                0443  ;  OUTPUT
                                0443  ;       R0 - IF LBS, LKB CONTAINS ADDRESS OF LOCK BLOCK
                                0443  ;            IF LBC, NO LOCK BLOCK FOR THIS ID
                                0443  ;
                                0443  ;---
                          000C  0443          .ENTRY  GET_LKB,^M<R2,R3>
                                0445
                                0445  ;
                                0445  ;  VALIDATE INPUT
                                0445  ;
52    00000090'EF       3C      0445          MOVZWL  LOCKID,R2               ; MOVE TO REGISTER AND TEST
                43      13      044C          BEQL    20$                     ; IF EQL, NOT VALID
                                044E          REQMEM  @LCK$GL_MAXID,R3        ; GET MAX ID VALUE
        53      52      D1      045E          CMPL    R2,R3                   ; CHECK IT
                41      1A      0461          BGTRU   30$                     ; IF GTRU, TOO BIG
                                0463
                                0463  ;
                                0463  ;  GET ADDRESS OF LOCK BLOCK
                                0463  ;
                                0463          REQMEM  @LCK$GL_IDTBL,R3        ; GET START OF TABLE
        53    6342      DE      0473          MOVAL   (R3)[R2],R3             ; CALC SLOT ADDRESS
                                0477          REQMEM  (R3),R3                 ; GET LKB ADDRESS
                50      D4      0483          CLRL    R0                      ; ASSUME EMPTY SLOT
000002BF'EF     53      D0      0485          MOVL    R3,LKB                  ; MOVE DATA AND TEST
                02      18      048C          BGEQ    10$                     ; IF GEQ, NO LKB FOR THIS ID
                50      D6      048E          INCL    R0                      ; INDICATE SUCCESS
                                0490
                        04      0490  10$:    RET                             ; DONE
                                0491
                                0491  20$:    SIGNAL  0,LOCKIDZER             ; LOCK ID 0
                        04      04A3          RET
                                04A4
                                04A4  30$:    SIGNAL  0,OUTOFRANG             ; LOCK ID GREATER THAN MAXID
                        04      04B6          RET
```

```
                           04B7        .SBTTL  SAVE_LOCK_DATA - RETRIEVE LOCK DATA FROM SYSTEM
                           04B7  ;+++
                           04B7  ;
                           04B7  ;       SAVE_LOCK_DATA
                           04B7  ;
                           04B7  ;  PURPOSE
                           04B7  ;       GIVEN ADDRESS OF LOCK BLOCK, RETRIEVE ALL DATA NEEDED TO
                           04B7  ;       DISPLAY LOCK INFORMATION
                           04B7  ;
                           04B7  ;  INPUT
                           04B7  ;       LKB - ADDRESS OF LOCK BLOCK
                           04B7  ;
                           04B7  ;  OUTPUT
                           04B7  ;       LKB_BFR     - CONTENTS OF LOCK BLOCK
                           04B7  ;       PARID_BFR   - ID OF PARENT LOCK
                           04B7  ;       LKB_RSB_BFR - DATA FROM ASSOCIATED RESOURCE
                           04B7  ;
                           04B7  ;---
                     000C  04B7        .ENTRY  SAVE_LOCK_DATA,^M<R2,R3>
                           04B9
52   000002BF'EF      D0   04B9        MOVL    LKB,R2
                           04C0        REQMEM  (R2),LKB_BFR,#LKB$K_LENGTH        ; STORE LKB DATA
53   00000050'EF      D0   04D5        MOVL    LKB_BFR+LKB$L_PARENT,R3           ; PARENT LKB ADDR
               0D     13   04DC        BEQL    10$                               ; IF EQL, NO PARENT
                           04DE        REQMEM  LKB$L_LKID(R3),R3                 ; FETCH PARENT LOCK ID
                           04EB
00000004'EF    53     D0   04EB  10$:  MOVL    R3,PARID_BFR                      ; STORE IN BUFFER
53   00000058'EF      D0   04F2        MOVL    LKB_BFR+LKB$L_RSB,R3              ; GET ASSOCIATED RSB
                           04F9
                           04F9        ASSUME  RSB$B_RMOD-RSB$W_GROUP    EQ 2  ; MAKE SURE NOTHING CHANGES
                           04F9        ASSUME  RSB$B_RSNLEN-RSB$W_GROUP  EQ 3
                           04F9        ASSUME  RSB$T_RESNAM-RSB$W_GROUP  EQ 4
                           04F9        ASSUME  RSB$K_MAXLEN EQ <^X1F>
                           04F9
                           04F9        REQMEM  RSB$W_GROUP(R3),LKB_RSB_BFR,-   ; STORE RSB DATA
                           04F9                 #RSB$K_MAXLEN+4
                           050B        REQMEM  RSB$L_CSID(R3),RSBCSID_BFR      ; GET CSID IN RSB
                           051C
                     04    051C        RET
```

```
                                 051D             .SBTTL  DISPLAY_LOCK - CONTROL DISPLAY OF LOCK DATA
                                 051D   ;+++
                                 051D   ;
                                 051D   ;         DISPLAY_LOCK
                                 051D   ;
                                 051D   ;   PURPOSE
                                 051D   ;         CONTROLS FORMAT AND DISPLAY OF LOCK DATA
                                 051D   ;
                                 051D   ;   INPUT
                                 051D   ;         LOCK DATA AREAS (IMPLIED)
                                 051D   ;
                                 051D   ;   OUTPUT
                                 051D   ;         DISPLAYED LOCK INFORMATION
                                 051D   ;
                                 051D   ;---
                           0000  051D             .ENTRY  DISPLAY_LOCK,^M<>
                                 051F
    00000556'EF     00     FB    051F             CALLS   #0,FORMAT_LOCK           ; FORMAT LOCK DATA
                                 0526             SKIP    PAGE
                                 052D             ENSURE  10
                                 0545             SKIP    1
    00000856'EF     00     FB    054E             CALLS   #0,PRINT_LOCK            ; DISPLAY LOCK DATA
                                 0555
                             04  0555             RET
```

```
                                     0556          .SBTTL  FORMAT_LOCK - FORMAT LOCK DATA
                                     0556  ;+++
                                     0556  ;
                                     0556  ;       FORMAT_LOCK
                                     0556  ;
                                     0556  ;  PURPOSE
                                     0556  ;       FORMAT RAW SYSTEM LOCK AND RESOURCE DATA. STORE
                                     0556  ;       FOR EASY ACCESS.
                                     0556  ;
                                     0556  ;  INPUT
                                     0556  ;       LKB_BFR - ADDRESS OF LOCK BLOCK DATA
                                     0556  ;       PARID_BFR - PARENT LOCK ID
                                     0556  ;       LKB_RSB_BFR - ASSOCIATED RESOURCE DATA
                                     0556  ;
                                     0556  ;  OUTPUT
                                     0556  ;       STORED DATA
                                     0556  ;
                                     0556  ;---
                                     0556
                               007C  0556          .ENTRY  FORMAT_LOCK,^M<R2,R3,R4,R5,R6>
                                     0558
                                     0558          ; Lock id., parent id., sublocks and PID
                                     0558
          56   00000008'EF     9E    0558          MOVAB   LKB_BFR,R6
     0000026F'EF     30 A6     D0    055F          MOVL    LKB$L_LKID(R6),LKID
 00000283'EF   00000004'EF     D0    0567          MOVL    PARID_BFR,PARID
     0000029F'EF     4C A6     3C    0572          MOVZWL  LKB$W_REFCNT(R6),SUBLKS
     00000273'EF     0C A6     D0    057A          MOVL    LKB$L_PID(R6),PID
                                     0582
                                     0582          ; Lock state and mode(s)
                                     0582
                     36 A6     95    0582          TSTB    LKB$B_STATE(R6)
                        37     19    0585          BLSS    20$                       ; Waiting
                        26     14    0587          BGTR    10$                       ; Granted
     00000287'EF   FC79 CF     9E    0589          MOVAB   CV_STRING,STATE1
     000002A3'EF   FC65 CF     9E    0592          MOVAB   GR_STRING,STATE2
              50     35 A6     9A    059B          MOVZBL  LKB$B_GRMODE(R6),R0       ; Get granted mode
     000002AB'EF   FC21 CF40   3E    059F          MOVAW   LOCKMODE_TBL[R0],MODE2    ; Store granted mode sting
              50     34 A6     9A    05A9          MOVZBL  LKB$B_RQMODE(R6),R0       ; Get requested mode
                        2E     11    05AD          BRB     30$
     00000287'EF   FC48 CF     9E    05AF  10$:    MOVAB   GR_STRING,STATE1
              50     35 A6     9A    05B8          MOVZBL  LKB$B_GRMODE(R6),R0
                        0D     11    05BC          BRB     25$
     00000287'EF   FC52 CF     9E    05BE  20$:    MOVAB   WT_STRING,STATE1
              50     34 A6     9A    05C7          MOVZBL  LKB$B_RQMODE(R6),R0
     000002A3'EF   FC53 CF     9E    05CB  25$:    MOVAB   NULL_CSTRING,STATE2       ; No state 2
     000002AB'EF   FC48 CF     9E    05D4          MOVAB   BLANKS,MODE2              ; No mode 2
     0000028F'EF   FBE3 CF40   3E    05DD  30$:    MOVAW   LOCKMODE_TBL[R0],MODE1    ; Store mode 1
                                     05E7
                                     05E7          ; Blocking AST
                                     05E7
     000002BB'EF   FC37 CF     9E    05E7          MOVAB   NULL_CSTRING,BLKAST       ; Assume no blocking AST
                     20 A6     D5    05F0          TSTL    LKB$L_BLKASTADR(R6)       ; Is there a blocking AST?
                        09     13    05F3          BEQL    32$
     000002BB'EF   FC2A CF     9E    05F5          MOVAB   BL_STRING,BLKAST          ; Yes, store address of string
                                     05FE
                                     05FE          ; Flags
```

```
                                05FE
       53    FC20 CF     9E     05FE  32$:    MOVAB   NULL_CSTRING,R3          ; Initialize fields
 00000277'EF      53     D0     0603          MOVL    R3,F[AGS1
 0000027B'EF      53     D0     060A          MOVL    R3,FLAGS1+4
 0000027F'EF      53     D0     0611          MOVL    R3,FLAGS1+8
 00000293'EF      53     D0     0618          MOVL    R3,FLAGS2
 00000297'EF      53     D0     061F          MOVL    R3,FLAGS2+4
 0000029B'EF      53     D0     0626          MOVL    R3,FLAGS2+8
 000002AF'EF      53     D0     062D          MOVL    R3,FLAGS3
 000002B3'EF      53     D0     0634          MOVL    R3,FLAGS3+4
 000002B7'EF      53     D0     063B          MOVL    R3,FLAGS3+8
                                0642
       55      28 A6     3C     0642          MOVZWL  LKB$W_FLAGS(R6),R5       ; Pick up flags
   54   00000277'EF      DE     0646          MOVAL   FLAGST,R4                ; Address of 1st arg. list
                  52     D4     064D          CLRL    R2
                  53     D4     064F          CLRL    R3
       25 55      53     E1     0651  35$:    BBC     R3,R5,60$                ; Br. if flag not set
   64    FBD0 CF43       7E     0655          MOVAQ   FLAGS_TBL[R3],(R4)       ; Store appropriate string address
          54      04     C0     065B          ADDL    #4,R4
                  52     D6     065E          INCL    R2                       ; Incr. position counter
          03      52     D1     0660          CMPL    R2,#3                    ; Move to 2nd line?
                  09     12     0663          BNEQ    40$                      ; No
   54   00000293'EF      DE     0665          MOVAL   FLAGS2,R4                ; Yes
                  0C     11     066C          BRB     60$
          06      52     D1     066E  40$:    CMPL    R2,#6                    ; Move to third line
                  07     12     0671          BNEQ    60$                      ; No
   54   000002AF'EF      DE     0673          MOVAL   FLAGS3,R4                ; Yes
       D3 53      09     F2     067A  60$:    AOBLSS  #9,R3,35$                ; Repeat 9 times
                                067E
                                067E
                                067E  PART2:
                                067E
                                067E          ; Status
                                067E
       53    FBA0 CF     9E     067E          MOVAB   NULL_CSTRING,R3          ; Initialize fields
 000002D3'EF      53     D0     0683          MOVL    R3,STS1
 000002D7'EF      53     D0     068A          MOVL    R3,STS1+4
 000002DB'EF      53     D0     0691          MOVL    R3,STS1+8
 000002F3'EF      53     D0     0698          MOVL    R3,STS2
 000002F7'EF      53     D0     069F          MOVL    R3,STS2+4
 000002FB'EF      53     D0     06A6          MOVL    R3,STS2+8
 00000313'EF      53     D0     06AD          MOVL    R3,STS3
 00000317'EF      53     D0     06B4          MOVL    R3,STS3+4
 0000031B'EF      53     D0     06BB          MOVL    R3,STS3+8
                                06C2
       55      2A A6     3C     06C2          MOVZWL  LKB$W_STATUS(R6),R5      ; Pick up status
   54   000002D3'EF      DE     06C6          MOVAL   STS1,R4                  ; Address of 1st arg. list
                  52     D4     06CD          CLRL    R2
                  53     D4     06CF          CLRL    R3
       25 55      53     E1     06D1  35$:    BBC     R3,R5,60$                ; Br. if flag not set
   64    FB98 CF43       7E     06D5          MOVAQ   STATUS_TBL[R3],(R4)      ; Store appropriate string address
          54      04     C0     06DB          ADDL    #4,R4
                  52     D6     06DE          INCL    R2                       ; Incr. position counter
          03      52     D1     06E0          CMPL    R2,#3                    ; Move to 2nd line?
                  09     12     06E3          BNEQ    40$                      ; No
   54   000002F3'EF      DE     06E5          MOVAL   STS2,R4                  ; Yes
                  0C     11     06EC          BRB     60$
```

```
             06    52    D1  06EE   797 40$:    CMPL    R2,#6                        ; Move to third line
                   07    12  06F1   798         BNEQ    60$                          ; No
      54    00000313'EF  DE  06F3   799         MOVAL   STS3,R4                      ; Yes
            D3 53    09  F2  06FA   800 60$:    AOBLSS  #9,R3,35$                    ; Repeat 9 times
                             06FE   801
                             06FE   802         ;
                             06FE   803         ; Resource name, length, access mode, and name space
                             06FE   804         ;
                             06FE   805
000002DF'EF    0000006F'EF   9A  06FE   806     MOVZBL  RSNLEN_BFR,RLEN
         51    0000006F'EF   9A  0709   807     MOVZBL  RSNLEN_BFR,R1
         50    00000070'EF   9E  0710   808     MOVAB   RESNAM_BFR,R0
                        50   DD  0717   809     PUSHL   R0                           ; Save R0
   60   20    00    60  51   2C  0719   810     MOVC5   R1,(R0),#0,#32,(R0)          ; Zero out rest of buffer
                     50 8ED0     071F   811     POPL    R0                           ; Restore R0
        000002CF'EF     60   DE  0722   812     MOVAL   (R0),DMP1
        000002C7'EF     80   D0  0729   813     MOVL    (R0)+,RES1
        000002C3'EF     80   D0  0730   814     MOVL    (R0)+,RES2
        000002EF'EF     60   DE  0737   815     MOVAL   (R0),DMP3
        000002E7'EF     80   D0  073E   816     MOVL    (R0)+,RES3
        000002E3'EF     80   D0  0745   817     MOVL    (R0)+,RES4
        0000030F'EF     60   DE  074C   818     MOVAL   (R0),DMP5
        00000307'EF     80   D0  0753   819     MOVL    (R0)+,RES5
        00000303'EF     80   D0  075A   820     MOVL    (R0)+,RES6
        0000032F'EF     60   DE  0761   821     MOVAL   (R0),DMP7
        00000327'EF     80   D0  0768   822     MOVL    (R0)+,RES7
        00000323'EF     80   D0  076F   823     MOVL    (R0)+,RES8
        50    0000006E'EF    9A  0776   824     MOVZBL  RMOD_BFR,R0                  ; Get access mode
  000002FF'EF    FB96 CF40   D0  077D   825     MOVL    ACMODE_TBL[R0],ACMODE
              000001BA'EF    B5  0787   826     TSTW    GROUP_BUF                    ; System names have group = 0
                        44   13  078D   827     BEQL    70$                          ; Branch if system name
                             078F   828         $FAO_S  CTRSTR = FAO_GROUP_DSC,-
                             078F   829                 OUTBUF = GROUP_BUF_DSC,-
                             078F   830                 P1 =  GROUP_BFR
                     03 50   E8  07AA   831     BLBS    R0,65$
                      00A5   31  07AD   832     BRW     90$
000001E8'EF    000001BA'EF   B0  07B0   833 65$:    MOVW    GROUP_BUF,GROUP_NUM
000001EA'EF    000001BC'EF   90  07BB   834         MOVB    GROUP_BUF+2,GROUP_NUM+2
0000031F'EF    000001D8'EF   7E  07C6   835         MOVAQ   GROUP_DSC,NSPACE
                        0B   11  07D1   836         BRB     80$
0000031F'EF    000001C8'EF   7E  07D3   837 70$:    MOVAQ   SYS_DSC,NSPACE
                             07DE   838
                             07DE   839 80$:    ; Format type of lock and remote lock info.
                             07DE   840
         50    00000068'EF   D0  07DE   841     MOVL    RSBCSID_BFR,R0               ; Get CSID
                        19   12  07E5   842     BNEQ    82$                          ; Must be process copy
            1F  2A A6   04   E0  07E7   843     BBS     #LKB$V_MSTCPY,LKB$W_STATUS(R6),84$ ; Branch if master copy
   00000333'EF    FACA CF    9E  07EC   844     MOVAB   LOCAL,TYPE                   ; Set type of lock
   00000337'EF    FAD6 CF    9E  07F5   845     MOVAB   SPACE,RMINFO                 ; Display no remote info
                        52   11  07FE   846     BRB     88$
                             0800   847
   00000333'EF    FABC CF    9E  0800   848 82$:    MOVAB   PROCESS,TYPE                 ; Set type of lock
                        0D   11  0809   849         BRB     86$
                             080B   850
   00000333'EF    FAB9 CF    9E  080B   851 84$:    MOVAB   MASTER,TYPE                  ; Set type of lock
          50     58 A6       D0  0814   852         MOVL    LKB$L_CSID(R6),R0            ; Get CSID
                             0818   853
```

```
                    51   54 A6   D0  0818   854 86$:    MOVL    LKB$L_REMLKID(R6),R1    ; Get remote lock id.
                                     081C   855         $FAO_S  CTRSTR = FAO_RMINFO_DSC,-
                                     081C   856                 OUTBUF = RMINFO_DSC,-
                                     081C   857                 OUTLEN = RMINFO_LEN,-
                                     081C   858                 P1 = R1,-
                                     081C   859                 P2 = R0
                         19 50   E9  0839   860         BLBC    R0,90$
   000000C4'EF   000000B8'EF     90  083C   861         MOVB    RMINFO_LEN,RMINFO_CNT
   00000337'EF   000000C4'EF     9E  0847   862         MOVAB   RMINFO_CNT,RMINFO       ; Store ptr to .ASCIC string
                                     0852   863
                      50    01   D0  0852   864 88$:    MOVL    #1,R0
                                 04  0855   865 90$:    RET
                         0856            .SBTTL  PRINT_LOCK - OUTPUT LOCK DATA
                         0856    ;+++
                         0856    ;
                         0856    ;       PRINT_LOCK
                         0856    ;
                         0856    ;  PURPOSE
                         0856    ;       OUTPUT FORMATTED LOCK DATA, LINE-BY-LINE.
                         0856    ;
                         0856    ;  INPUT
                         0856    ;       FORMATTED LOCK DATA
                         0856    ;
                         0856    ;  OUTPUT
                         0856    ;       DISPLAYED DATA
                         0856    ;
                         0856    ;---
                 0000    0856            .ENTRY  PRINT_LOCK,^M<>
                         0858
                         0858    ;
                         0858    ;  LINE 1
                         0858    ;
       0000027F'EF   DD  0858            PUSHL   FLAGS1+8
       0000027B'EF   DD  085E            PUSHL   FLAGS1+4
       00000277'EF   DD  0864            PUSHL   FLAGS1
       00000273'EF   DD  086A            PUSHL   PID
       0000026F'EF   DD  0870            PUSHL   LKID
                         0876            PRINT   5,<Lock id:  !XL   PID:     !XL   Flags:    !3(8AC)>
                         0883    ;                            LKID             PID                FLAGS1
                         0883
                         0883    ;
                         0883    ;  LINE 2
                         0883    ;
       0000029B'EF   DD  0883            PUSHL   FLAGS2+8
       00000297'EF   DD  0889            PUSHL   FLAGS2+4
       00000293'EF   DD  088F            PUSHL   FLAGS2
       0000028F'EF   DD  0895            PUSHL   MODE1
                02   DD  089B            PUSHL   #2
       00000287'EF   DD  089D            PUSHL   STATE1
       00000283'EF   DD  08A3            PUSHL   PARID
                         08A9            PRINT   7,<Par. id:  !XL   !13AC !AD!12* !3(8AC)>
                         08B6    ;                          PARID STATE1 MODE1    FLAGS2
                         08B6
                         08B6    ;
                         08B6    ;  LINE 3
                         08B6    ;
       000002B7'EF   DD  08B6            PUSHL   FLAGS3+8
       000002B3'EF   DD  08BC            PUSHL   FLAGS3+4
       000002AF'EF   DD  08C2            PUSHL   FLAGS3
       000002AB'EF   DD  08C8            PUSHL   MODE2
                02   DD  08CE            PUSHL   #2
       000002A3'EF   DD  08D0            PUSHL   STATE2
       0000029F'EF   DD  08D6            PUSHL   SUBLKS
                         08DC            PRINT   7,<Sublocks:  !7UL   !13AC !AD!12* !3(8AC)>
                         08E9    ;                            SUBLKS STATE2 MODE2    FLAGS3
                         08E9
                         08E9    ;
                         08E9    ;  LINE 3A
                         08E9    ;
```

```
000002BB'EF  DD  08E9         PUSHL   BLKAST
000002BF'EF  DD  08EF         PUSHL   LKB
                 08F5         PRINT   2,<LKB:        !XL   !9AC>
                 0902  ;                             LKB   BLKAST
                 0902
                 0902  ;
                 0902  ;  LINE 4
                 0902  ;
000002DB'EF  DD  0902         PUSHL   STS1+8
000002D7'EF  DD  0908         PUSHL   STS1+4
000002D3'EF  DD  090E         PUSHL   STS1
000002CF'EF  DD  0914         PUSHL   DMP1
         08  DD  091A         PUSHL   #8
000002C7'EF  DD  091C         PUSHL   RES1
000002C3'EF  DD  0922         PUSHL   RES2
                 0928         PRINT   7,<Resource:        !XL !XL      !AF  Status:  !3(8AC)>
                 0935  ;                                  RES2 RES1    DMP1            STS1
                 0935
                 0935  ;
                 0935  ;  LINE 5
                 0935  ;
000002FB'EF  DD  0935         PUSHL   STS2+8
000002F7'EF  DD  093B         PUSHL   STS2+4
000002F3'EF  DD  0941         PUSHL   STS2
000002EF'EF  DD  0947         PUSHL   DMP3
         08  DD  094D         PUSHL   #8
000002E7'EF  DD  094F         PUSHL   RES3
000002E3'EF  DD  0955         PUSHL   RES4
000002DF'EF  DD  095B         PUSHL   RLEN
                 0961         PRINT   8,< Length    !2ZB    !XL !XL    !AF                !3(8AC)>
                 096E  ;                            RLEN    RES4 RES3  DMP3                 STS2
                 096E
                 096E  ;
                 096E  ;  LINE 6
                 096E  ;
0000031B'EF  DD  096E         PUSHL   STS3+8
00000317'EF  DD  0974         PUSHL   STS3+4
00000313'EF  DD  097A         PUSHL   STS3
0000030F'EF  DD  0980         PUSHL   DMP5
         08  DD  0986         PUSHL   #8
00000307'EF  DD  0988         PUSHL   RES5
00000303'EF  DD  098E         PUSHL   RES6
000002FF'EF  DD  0994         PUSHL   ACMODE
                 099A         PRINT   8,< !11AC    !XL !XL    !AF             !3(8AC)>
                 09A7  ;                 ACMODE  RES6 RES5   DMP5              STS3
                 09A7
                 09A7  ;
                 09A7  ;  LINE 7
                 09A7  ;
0000032F'EF  DD  09A7         PUSHL   DMP7
         08  DD  09AD         PUSHL   #8
00000327'EF  DD  09AF         PUSHL   RES7
00000323'EF  DD  09B5         PUSHL   RES8
0000031F'EF  DD  09BB         PUSHL   NSPACE
                 09C1         PRINT   5,< !11AS   !XL !XL    !AF>
                 09CE  ;                 NSPACE  RES8 RES7   DMP7
                 09CE
```

```
                         09CE    981 ;
                         09CE    982 ; LINE 8
                         09CE    983 ;
      00000337'EF    DD  09CE    984          PUSHL   RMINFO
      00000333'EF    DD  09D4    985          PUSHL   TYPE
                         09DA    986          PRINT   2,<!AC copy !AC!/>
                         09E7    987 ;                    TYPE      RMINFO
                     04  09E7    988          RET
```

```
                                  09E8          .SBTTL  LOCK_COND_HAND - CONDITION HANDLER FOR SHOW_ALL_LOCKS
                                  09E8  ;+++
                                  09E8  ;
                                  09E8  ;         LOCK_COND_HAND
                                  09E8  ;
                                  09E8  ;   PURPOSE
                                  09E8  ;         PROVIDE EXIT PATH FOR SHOW_ALL_LOCKS WHEN THERE ARE NO MORE LOCKS
                                  09E8  ;         TO BE PROCESSED
                                  09E8  ;
                                  09E8  ;   INPUT
                                  09E8  ;         4(AP) = POINTER TO SIGNAL ARGUMENTS
                                  09E8  ;         8(AP) = POINTER TO MECHANISM ARGUMENTS
                                  09E8  ;
                                  09E8  ;   OUTPUT
                                  09E8  ;         POSSIBLE MODIFICATION OF STATUS; POSSIBLE CHANGE IN FLOW
                                  09E8  ;         OF CONTROL.
                                  09E8  ;
                                  09E8  ;---
                           001C   09E8          .ENTRY  LOCK_COND_HAND,^M<R2,R3,R4>
                                  09EA
            52     04 AC    7D    09EA          MOVQ    4(AP),R2                 ; GET ADDRESSES OF ARRAYS
            54     04 A2    D0    09EE          MOVL    CHF$L_SIG_NAME(R2),R4    ; GET CONDITION NAME
     54     00000000'8F     D1    09F2          CMPL    #SS$_UNWIND,R4           ; ARE WE UNWINDING?
                     31     13    09F9          BEQL    20$                      ; IF EQL, YES
     54     00000000'8F     D1    09FB          CMPL    #MSG$_OUTOFRANG,R4       ; DID WE GET THE SIGNAL WE'RE LOOKING FOR?
                     28     12    0A02          BNEQ    20$                      ; IF NEQ, NO - PASS IT ON
                                  0A04
            00000000'EF     D5    0A04          TSTL    LOCK_COUNT               ; DID WE PROCESS LOCKS?
                     12     12    0A0A          BNEQ    10$                      ; IF NEQ, YES
                                  0A0C          SIGNAL  0,NOLOCKS                ; DISPLAY INFO MESSAGE
                                  0A1E
          0C A3      50     D0    0A1E  10$:    MOVL    R0,CHF$L_MCH_SAVR0(R3)   ; INDICATE SUCCESS AFTER UNWIND
                     7E     7C    0A22          CLRQ    -(SP)                    ; GO BACK TO ESTABLISHER
     00000000'GF     02     FB    0A24          CALLS   #2,G^SYS$UNWIND          ; UNWIND CALL FRAMES
                            04    0A2B          RET                              ; RETURN TO ESTABLISHER
                                  0A2C
       50     0000'8F       3C    0A2C  20$:    MOVZWL  #SS$_RESIGNAL,R0         ; RESIGNAL CONDITION
                            04    0A31          RET
```

```
                                0A32  1029          .SBTTL  SHOW_ALL_RES - ACTION ROUTINE TO DISPLAY ALL RESOURCES
                                0A32  1030  ;+++
                                0A32  1031  ;
                                0A32  1032  ;       SHOW_ALL_RES
                                0A32  1033  ;
                                0A32  1034  ;   PURPOSE
                                0A32  1035  ;       MAIN ROUTINE TO PROCESS THE "SHOW RES/ALL" COMMAND
                                0A32  1036  ;
                                0A32  1037  ;   INPUT
                                0A32  1038  ;       NONE
                                0A32  1039  ;
                                0A32  1040  ;   OUTPUT
                                0A32  1041  ;       DISPLAYED DATA
                                0A32  1042  ;
                                0A32  1043  ;---
                        001C    0A32  1044          .ENTRY  SHOW_ALL_RES,^M<R2,R3,R4>
                                0A34  1045
             F5DD CF      9F    0A34  1046          PUSHAB  RES_HEAD                        ; SET UP HEADING
       00000000'EF  01    FB    0A38  1047          CALLS   #1,SET_HEADING
          00000115'EF     D4    0A3F  1048          CLRL    RES_COUNT                       ; INIT COUNT
                   54     D4    0A45  1049          CLRL    R4                              ; INIT HASH TABLE INDEX
                                0A47  1050          REQMEM  @LCK$GL_HTBLCNT,R2              ; GET COUNT OF ELEMENTS IN HASH TABLE
  00000190'EF   01  52    78    0A57  1051          ASHL    R2,#1,HTBL_CNT
          53  00000000'EF D0    0A5F  1052          MOVL    LCK$GL_HASHTBL,R3               ; GET ADDRESS OF START OF HASH TABLE
                                0A66  1053          REQMEM  (R3),R3                         ; GET FIRST ELEMENT
                                0A72  1054
                                0A72  1055  ;
                                0A72  1056  ;   INDEX THROUGH HASH TABLE, LOOKING FOR HASH CHAINS
                                0A72  1057  ;
                                0A72  1058  10$:
              52   6344   DE    0A72  1059          MOVAL   (R3)[R4],R2                     ; GET HEAD OF CHAIN
                                0A76  1060          REQMEM  (R2),RSB                        ; GET FIRST RSB ADDRESS
          000001EB'EF     D5    0A86  1061          TSTL    RSB                             ; IF ZERO, EMPTY CHAIN
                   3A     13    0A8C  1062          BEQL    20$
                                0A8E  1063
                                0A8E  1064  ;
                                0A8E  1065  ;   SEQUENCE THROUGH HASH CHAIN
                                0A8E  1066  ;
                                0A8E  1067  15$:
          00000115'EF     D6    0A8E  1068          INCL    RES_COUNT                       ; INCREMENT COUNT
                                0A94  1069          REQMEM  @RSB,RSB_BFR,#RSB$K_MAXLEN+RSB$K_LENGTH   ;SAVE RSB DATA
       00000B62'EF  00    FB    0AAD  1070          CALLS   #0,SHOW_RSB                     ; DISPLAY RSB
       00000B92'EF  00    FB    0AB4  1071          CALLS   #0,SHOW_QUEUES                  ; DISPLAY QUEUES
 000001EB'EF   0000011D'EF D0   0ABB  1072          MOVL    RSB_BFR,RSB                     ; GET NEXT RSB
                   C6     19    0AC6  1073          BLSS    15$
                                0AC8  1074
                                0AC8  1075  ;
                                0AC8  1076  ;   OUTER LOOP CONTROL
                                0AC8  1077  ;
                                0AC8  1078  20$:
     A2 54   00000190'EF  F2    0AC8  1079          AOBLSS  HTBL_CNT,R4,10$                 ; INCREMENT INDEX, LOOP IF LSS
                                0AD0  1080
          00000115'EF     D5    0AD0  1081          TSTL    RES_COUNT                       ; DID WE PROCESS ANY RESOURCES?
                   08     13    0AD6  1082          BEQL    30$                             ; IF EQL, NO
                                0AD8  1083
                                0AD8  1084          STATUS  SUCCESS                         ; SUCCESS RETURN
                   04           0ADF  1085          RET
```

```
LOC
Sym

$$T
ACM
ACM
ARG
BAS
BLA
BLK
BL_
BUF
BUF
BUF
BUF
CGM
CHF
CHF
CON
CON
CON
CON
CON
CON
COU
CRE
CSI
CV_
CWR
DIR
DIR
DIS
DMP
DMP
DMP
DMP
DON
EMO
EX_
EXI
FAO
FAO
FLA
FLA
FLA
FLA
FOR
FOR
GET
GET
GGM
GRA
GRA
GRM
GRM
GRO
GRO
GRO
GRO
GRO
```

```
        0AE0  1086 30$:
        0AE0  1087          SIGNAL  0,NORESOURC               ; NO RESOURCES DISPLAYED
    04  0AF2  1088          RET
```

```
LOC
Sym

NOT
NSP
NUL
NUL
PAG
PAR
PAR
PAR
PCB
PID
PRE
PRI
PRI
PRI
PRI
PRO
PRO
PRS
PWR
QUE
QUE
QUE
RAC
REQ
RES
RES
RES
RES
RES
RES
RES
RES
RES
RES
RES
RES
RES
RES
RES
RES
RES
RES
RES
RLE
RMI
RMI
RMI
RMI
RMI
RMC
RNS
RQM
RQM
RSE
RSE
RSE
```

```
                          0AF3
                          0AF3            .SBTTL  SHOW_ONE_RES - ACTION ROUTINE TO DISPLAY ONE RESOURCE
                          0AF3      ;+++
                          0AF3      ;
                          0AF3      ;         SHOW_ONE_RES
                          0AF3      ;
                          0AF3      ;   PURPOSE
                          0AF3      ;         MAIN ROUTINE TO PROCESS THE "SHOW RES/LOCK=lockid" COMMAND.
                          0AF3      ;
                          0AF3      ;   INPUT
                          0AF3      ;         LOCKID - ID OF LOCK WHOSE ASSOCIATED RESOURCE IS TO BE DISPLAYED.
                          0AF3      ;
                          0AF3      ;   OUTPUT
                          0AF3      ;         DISPLAYED DATA.
                          0AF3      ;
                          0AF3      ;---
                    0004  0AF3            .ENTRY  SHOW_ONE_RES,^M<R2>
                          0AF5
        F949 CF   00  FB  0AF5            CALLS   #0,GET_LKB                  ; GET LKB ADDRESS
              52 50   E9  0AFA            BLBC    R0,10$                      ; IF LBC, NONE EXISTS
52    000002BF'EF     D0  0AFD            MOVL    LKB,R2
                          0B04            REQMEM  LKB$L_RSB(R2),RSB           ; GET ADDRESS OF RSB
                          0B15            REQMEM  @RSB,RSB_BFR,#RSB$K_MAXLEN+RSB$K_LENGTH   ; GET RSB DATA
        F4E3 CF       9F  0B2E            PUSHAB  RES_HEAD                    ; SET RSB HEADING
00000000'EF       01  FB  0B32            CALLS   #1,SET_HEADING
00000B62'EF       00  FB  0B39            CALLS   #0,SHOW_RSB                 ; DISPLAY RSB
00000B92'EF       00  FB  0B40            CALLS   #0,SHOW_QUEUES              ; DISPLAY QUEUES
                          0B47
                          0B47            STATUS  SUCCESS
                      04  0B4E            RET
                          0B4F
                          0B4F      10$:  SIGNAL  0,NOLKB                     ; LOCKID NON-EXISTANT
                      04  0B61            RET
```

```
LOC
Pse

PSE
---
.
$AB
SDA
LOC
LIT

Pha
---
Ini
Com
Pas
Sym
Pas
Sym
Pse
Cro
Ass

The
666
The
149
26

Mac
---
_$2
-$2
-$2
TOT

660

The

MAC
```

```
                              0B62            .SBTTL  SHOW_RSB - CONTROL FOR RSB DISPLAY
                              0B62    ;+++
                              0B62    ;
                              0B62    ;       SHOW_RSB
                              0B62    ;
                              0B62    ; PURPOSE
                              0B62    ;       CONTROL THE FORMAT AND DISPLAY OF RSB DATA
                              0B62    ;
                              0B62    ; INPUT
                              0B62    ;       RSB DATA STRUCTURES (IMPLIED)
                              0B62    ;
                              0B62    ; OUTPUT
                              0B62    ;       DISPLAYED DATA
                              0B62    ;---
                        0000  0B62            .ENTRY  SHOW_RSB,^M<>
                              0B64
  00000BCC'EF    00     FB    0B64            CALLS   #0,FORMAT_RSB
                              0B6B            SKIP    PAGE
                              0B72            ENSURE  9
  00000D34'EF    00     FB    0B8A            CALLS   #0,PRINT_RSB
                              0B91
                          04  0B91            RET
```

```
                              0B92       .SBTTL  SHOW_QUEUES - DISPLAY QUEUES FOR GIVEN RESOURCE
                              0B92  ;+++
                              0B92  ;
                              0B92  ;        SHOW_QUEUES
                              0B92  ;
                              0B92  ;  PURPOSE
                              0B92  ;        DISPLAY, SEQUENTIALLY, THE GRANTED, CONVERSION AND WAIT
                              0B92  ;        QUEUES ASSOCIATED WITH A RESOURCE.
                              0B92  ;
                              0B92  ;  INPUT
                              0B92  ;        RSB - BUFFER CONTAINING RSB DATA
                              0B92  ;
                              0B92  ;  OUTPUT
                              0B92  ;        DISPLAYED QUEUES.
                              0B92  ;
                              0B92  ;---
                        0000  0B92           .ENTRY  SHOW_QUEUES,^M<>
                              0B94
                              0B94
7E   10   000001EB'EF    C1   0B94           ADDL3   RSB,#RSB$L_GRQFL,-(SP)          ; GET ADDRESS OF GRANT QUE FLINK
           FFFFFFFF 8F   DD   0B9C           PUSHL   #GRANT                          ; IND GRANT QUE
     00000E55'EF    02   FB   0BA2           CALLS   #2,PROCESS_QUEUE
                              0BA9
7E   18   000001EB'EF    C1   0BA9           ADDL3   RSB,#RSB$L_CVTQFL,-(SP)         ; GET ADDRESS OF CONVERT QUE FLINK
                    00   DD   0BB1           PUSHL   #CONVERT                        ; IND CONVERT QUE
     00000E55'EF    02   FB   0BB3           CALLS   #2,PROCESS_QUEUE
                              0BBA
7E   20   000001EB'EF    C1   0BBA           ADDL3   RSB,#RSB$L_WTQFL,-(SP)          ; GET ADDRESS OF WAIT QUE FLINK
                    01   DD   0BC2           PUSHL   #WAIT                           ; IND WAIT QUE
     00000E55'EF    02   FB   0BC4           CALLS   #2,PROCESS_QUEUE
                              0BCB
                        04    0BCB           RET
```

```
                                         0BCC  1180           .SBTTL  FORMAT_RSB - FORMAT RSB DATA
                                         0BCC  1181   ;+++
                                         0BCC  1182   ;
                                         0BCC  1183   ;       FORMAT_RSB
                                         0BCC  1184   ;
                                         0BCC  1185   ; PURPOSE
                                         0BCC  1186   ;       FORMAT RAW RSB DATA, STORE FOR EASY ACCESS LATER ON.
                                         0BCC  1187   ;
                                         0BCC  1188   ; INPUT
                                         0BCC  1189   ;       RSB_BFR - BUFFER CONTAINING RSB DATA
                                         0BCC  1190   ;
                                         0BCC  1191   ; OUTPUT
                                         0BCC  1192   ;       STORED DATA.
                                         0BCC  1193   ;
                                         0BCC  1194   ;---
                                   00FC  0BCC  1195           .ENTRY  FORMAT_RSB,^M<R2,R3,R4,R5,R6,R7>
                                         0BCE  1196
             56    0000011D'EF     9E    0BCE  1197           MOVAB   RSB_BFR,R6                          ; GET ADDRESS OF DATA BUFFER
                                         0BD5  1198
                                         0BD5  1199   ;
                                         0BD5  1200   ; FORMAT RESOURCE NAME
                                         0BD5  1201   ;
                    57      50 A6  9E    0BD5  1202           MOVAB   RSB$T_RESNAM(R6),R7                 ; LET R7 POINT TO RESOURCE NAME
                    50      4F A6  9A    0BD9  1203           MOVZBL  RSB$B_RSNLEN(R6),R0                 ; ZERO OUT UNUSED PORTION OF BUFFER
        67    20   00   67    50   2C    0BDD  1204           MOVC5   R0,(R7),#0,#32,(R7)
              00000227'EF   67     DE    0BE3  1205           MOVAL   (R7),BUF1
              0000021F'EF   87     D0    0BEA  1206           MOVL    (R7)+,RESN1
              0000021B'EF   87     D0    0BF1  1207           MOVL    (R7)+,RESN2
              0000023B'EF   67     DE    0BF8  1208           MOVAL   (R7),BUF2
              00000233'EF   87     D0    0BFF  1209           MOVL    (R7)+,RESN3
              0000022F'EF   87     D0    0C06  1210           MOVL    (R7)+,RESN4
              00000253'EF   67     DE    0C0D  1211           MOVAL   (R7),BUF3
              0000024B'EF   87     D0    0C14  1212           MOVL    (R7)+,RESN5
              00000247'EF   87     D0    0C1B  1213           MOVL    (R7)+,RESN6
              0000026B'EF   67     DE    0C22  1214           MOVAL   (R7),BUF4
              00000263'EF   87     D0    0C29  1215           MOVL    (R7)+,RESN7
              0000025F'EF   87     D0    0C30  1216           MOVL    (R7)+,RESN8
                                         0C37  1217
                                         0C37  1218   ;
                                         0C37  1219   ; FORMAT RSB
                                         0C37  1220   ;
         000001F3'EF        48 A6  D0    0C37  1221           MOVL    RSB$L_PARENT(R6),PRSB
         000001FB'EF        40 A6  3C    0C3F  1222           MOVZWL  RSB$W_REFCNT(R6),SRSBCT
         000001FF'EF        42 A6  3C    0C47  1223           MOVZWL  RSB$W_BLKASTCNT(R6),BAST
         0000020F'EF        28 A6  D0    0C4F  1224           MOVL    RSB$Q_VALBLK(R6),VAL1
         0000020B'EF        2C A6  D0    0C57  1225           MOVL    RSB$Q_VALBLK+4(R6),VAL2
         00000207'EF        30 A6  D0    0C5F  1226           MOVL    RSB$Q_VALBLK+8(R6),VAL3
         00000203'EF        34 A6  D0    0C67  1227           MOVL    RSB$Q_VALBLK+12(R6),VAL4
         00000213'EF        3C A6  D0    0C6F  1228           MOVL    RSB$L_VALSEQNUM(R6),SEQNUM
          0000023F'EF       38 A6  D0    0C77  1229           MOVL    RSB$L_CSID(R6),CSID
       00000217'EF    F64C  CF     9E    0C7F  1230           MOVAB   SPACE,VALID
                09 0E A6    01     E1    0C88  1231           BBC     #RSB$V_VALINVLD,RSB$W_STATUS(R6),15$
       00000217'EF    F640  CF     9E    0C8D  1232           MOVAB   NOT_VALID,VALID
       00000257'EF    F635  CF     9E    0C96  1233   15$:    MOVAB   SPACE,DIRENTRY
                09 0E A6    00     E1    0C9F  1234           BBC     #RSB$V_DIRENTRY,RSB$W_STATUS(R6),17$
       00000257'EF    F633  CF     9E    0CA4  1235           MOVAB   DIR_ENTRY,DIRENTRY
                   51    0C A6     9A    0CAD  1236   17$:    MOVZBL  RSB$B_GGMODE(R6),R1
        000001EF'EF     F51B CF41   D0  0CB1   1237          MOVL    LKMODE_TBL[R1],GGMOD
                      51   0D A6   9A  0CBB   1238          MOVZBL  RSB$B_CGMODE(R6),R1
        000001F7'EF     F50D CF41   D0  0CBF   1239          MOVL    LKMODE_TBL[R1],CGMOD
                      51   4E A6   90  0CC9   1240          MOVB    RSB$B_RMOD(R6),R1
        00000243'EF     F646 CF41   D0  0CCD   1241          MOVL    ACMODE_TBL[R1],RACMOD
            0000022B'EF     4F A6   9A  0CD7   1242          MOVZBL  RSB$B_RSNLEN(R6),RESNLEN
      0000025B'EF    000001C8'EF   7E  0CDF   1243          MOVAQ   SYS_DSC,RNSPACE
            0000006C'EF     4C A6   B0  0CEA   1244          MOVW    RSB$W_GROUP(R6),GROUP_BFR
                               3F   13  0CF2   1245          BEQL    90$                               ; Branch if system name
                                        0CF4   1246          $FAO_S  CTRSTR = FAO_GROUP_DSC,-
                                        0CF4   1247                  OUTBUF = GROUP_BUF_DSC,-
                                        0CF4   1248                  P1 =  GROUP_BFR
                            21 50   E9  0D0F   1249          BLBC    R0,90$
      000001E8'EF    000001BA'EF   B0  0D12   1250          MOVW    GROUP_BUF,GROUP_NUM
      000001EA'EF    000001BC'EF   90  0D1D   1251          MOVB    GROUP_BUF+2,GROUP_NUM+2
      0000025B'EF    000001D8'EF   7E  0D28   1252          MOVAQ   GROUP_DSC,RNSPACE
                                        0D33   1253
                                        0D33   1254 90$:
                                    04  0D33   1255          RET
```

```
                  0D34          .SBTTL  PRINT_RSB - OUTPUT RSB DATA
                  0D34  ;+++
                  0D34  ;
                  0D34  ;       PRINT_RSB
                  0D34  ;
                  0D34  ;  PURPOSE
                  0D34  ;       OUTPUT RSB DATA, LINE-BY-LINE.
                  0D34  ;
                  0D34  ;  INPUT
                  0D34  ;       FORMATTED DATA.
                  0D34  ;
                  0D34  ;  OUTPUT
                  0D34  ;       DIPLAYED DATA.
                  0D34  ;
                  0D34  ;---
                  0D34
            0000  0D34          .ENTRY  PRINT_RSB,^M<>
                  0D36
                  0D36  ;
                  0D36  ;  LINE 1
                  0D36  ;
000001EF'EF   DD  0D36          PUSHL   GGMOD
000001EB'EF   DD  0D3C          PUSHL   RSB
                  0D42          PRINT   2,<Address of RSB:  !XL  Group grant mode:       !AC      >
                                                            RSB                         GGMOD
                  0D4F  ;
                  0D4F
                  0D4F  ;
                  0D4F  ;  LINE 2
                  0D4F  ;
000001F7'EF   DD  0D4F          PUSHL   CGMOD
000001F3'EF   DD  0D55          PUSHL   PRSB
                  0D5B          PRINT   2,<Parent RSB:      !XL  Conversion grant mode:  !AC      >
                                                            PRSB                         CGMOD
                  0D68  ;
                  0D68
                  0D68  ;
                  0D68  ;  LINE 3
                  0D68  ;
000001FF'EF   DD  0D68          PUSHL   BAST
000001FB'EF   DD  0D6E          PUSHL   SRSBCT
                  0D74          PRINT   2,<Sub-RSB count:   !8UL  BLKAST count:     !8UL>
                                                           SRSBCT                   BAST
                  0D81  ;
                  0D81
                  0D81  ;
                  0D81  ;  LINE 4
                  0D81  ;
00000217'EF   DD  0D81          PUSHL   VALID
00000213'EF   DD  0D87          PUSHL   SEQNUM
0000020F'EF   DD  0D8D          PUSHL   VAL1
0000020B'EF   DD  0D93          PUSHL   VAL2
00000207'EF   DD  0D99          PUSHL   VAL3
00000203'EF   DD  0D9F          PUSHL   VAL4
                  0DA5          PRINT   6,<Value block:   !XL !XL !XL !XL   Seq. #:  !XL !AC>
                                                        VAL4 VAL3 VAL2 VAL1       SEQNUM VALID
                  0DB2  ;
                  0DB2
                  0DB2  ;
                  0DB2  ;  LINE 5
                  0DB2  ;
```

```
00000227'EF  DD  0DB2          PUSHL   BUF1
         08  DD  0DB8          PUSHL   #8
0000021F'EF  DD  0DBA          PUSHL   RESN1
0000021B'EF  DD  0DC0          PUSHL   RESN2
                 0DC6          PRINT   4,<Resource:        !XL !XL      !AF  >
                 0DD3    ;                                    RESN2 RESN1  BUF1
                 0DD3
                 0DD3    ;
                 0DD3    ;  LINE 6
                 0DD3    ;
0000023F'EF  DD  0DD3          PUSHL   CSID
0000023B'EF  DD  0DD9          PUSHL   BUF2
         08  DD  0DDF          PUSHL   #8
00000233'EF  DD  0DE1          PUSHL   RESN3
0000022F'EF  DD  0DE7          PUSHL   RESN4
0000022B'EF  DD  0DED          PUSHL   RESNLEN
                 0DF3          PRINT   6,< Length    !2UL   !XL !XL      !AF           CSID:  !XL>
                 0E00    ;                         RESNLEN RESN4 RESN3  BUF2                  CSID
                 0E00
                 0E00    ;
                 0E00    ;  LINE 7
                 0E00    ;
00000257'EF  DD  0E00          PUSHL   DIRENTRY
00000253'EF  DD  0E06          PUSHL   BUF3
         08  DD  0E0C          PUSHL   #8
0000024B'EF  DD  0E0E          PUSHL   RESN5
00000247'EF  DD  0E14          PUSHL   RESN6
00000243'EF  DD  0E1A          PUSHL   RACMOD
                 0E20          PRINT   6,< !11AC    !XL !XL      !AF           !AC>
                 0E2D    ;                 RACMOD RESN6 RESN5  BUF3        DIRENTRY
                 0E2D
                 0E2D    ;
                 0E2D    ;  LINE 8
                 0E2D    ;
0000026B'EF  DD  0E2D          PUSHL   BUF4
         08  DD  0E33          PUSHL   #8
00000263'EF  DD  0E35          PUSHL   RESN7
0000025F'EF  DD  0E3B          PUSHL   RESN8
0000025B'EF  DD  0E41          PUSHL   RNSPACE
                 0E47          PRINT   5,< !11AS    !XL !XL      !AF>
                 0E54    ;                         RNSPACE RESN8 RESN7 BUF4
                 0E54
         04      0E54          RET
```

```
                              0E55  1358          .SBTTL  PROCESS_QUEUE - TRAVERSE RESOURCE QUEUES
                              0E55  1359  ;+++
                              0E55  1360  ;
                              0E55  1361  ;       PROCESS_QUEUE
                              0E55  1362  ;
                              0E55  1363  ;  PURPOSE
                              0E55  1364  ;       TRAVERSE AN INDIVIDUAL RESOURCE QUEUE, FORMATTING AND
                              0E55  1365  ;       DISPLAYING ITS CONTENTS.
                              0E55  1366  ;
                              0E55  1367  ;  INPUT
                              0E55  1368  ;       4(AP) - QUEUE TYPE, WHERE:  -1: GRANT
                              0E55  1369  ;                                    0: CONVERSION
                              0E55  1370  ;                                    1: WAIT
                              0E55  1371  ;       8(AP) - ADDRESS OF HEAD OF QUEUE
                              0E55  1372  ;
                              0E55  1373  ;  OUTPUT
                              0E55  1374  ;       DISPLAYED CONTENTS OF QUEUE
                              0E55  1375  ;
                              0E55  1376  ;---
                        03FC  0E55  1377          .ENTRY  PROCESS_QUEUE,^M<R2,R3,R4,R5,R6,R7,R8,R9>
                              0E57  1378
      54  000001A2'EF    9E  0E57  1379          MOVAB   LKID_BFR,R4             ; GET ADDRESS OF LOCKID STORAGE BUFFER
      55  0000019C'EF    9E  0E5E  1380          MOVAB   GRMD_BFR,R5             ; GET ADDRESS OF GRANT MODE BUFFER
      56  0000019F'EF    9E  0E65  1381          MOVAB   RQMD_BFR,R6             ; GET ADDRESS OF REQUEST MODE BUFFER
             52   08 AC  D0  0E6C  1382          MOVL    8(AP),R2                ; GET ADDRESS OF HEAD OF QUEUE
                59   52  D0  0E70  1383          MOVL    R2,R9
          00000119'EF    D4  0E73  1384          CLRL    QUEUE_COUNT             ; INIT COUNT OF QUEUE ITEMS PROCESSED
 00000198'EF      04 AC  D0  0E79  1385          MOVL    4(AP),QUEUE_TYPE        ; GET TYPE OF QUEUE
                58   01  CE  0E81  1386          MNEGL   #1,R8
   57  00000198'EF   01  C1  0E84  1387          ADDL3   #1,QUEUE_TYPE,R7
        57   F1BD CF47   D0  0E8C  1388          MOVL    QUE_STR_TBL[R7],R7      ; GET HEADER FOR THIS TYPE OF QUEUE
                             0E92  1389          ENSURE  3
                             0EAA  1390          SKIP    1
                             0EB3  1391          PRINTD  R8,(R7)                 ; DISPLAY IT
                             0EBE  1392
                             0EBE  1393  ;
                             0EBE  1394  ;  INIT FOR SCAN OF QUEUE
                             0EBE  1395  ;
                     53  D4  0EBE  1396  10$:    CLRL    R3
                             0EC0  1397
                             0EC0  1398  ;
                             0EC0  1399  ;  GET QUEUE ELEMENT
                             0EC0  1400  ;
                             0EC0  1401  20$:    REQMEM  (R2),R2                 ; GET QUEUE ELEMENT
              52     59  D1  0ECC  1402          CMPL    R9,R2                   ; AT END OF QUEUE?
                     3D  13  0ECF  1403          BEQL    30$                     ; IF EQL, YES
                             0ED1  1404
                             0ED1  1405          ASSUME  LKB$L_SQFL-LKB$L_LKID   GT  0
                             0ED1  1406          ASSUME  LKB$B_RQMODE-LKB$L_LKID EQ  4
                             0ED1  1407          ASSUME  LKB$B_GRMODE-LKB$L_LKID EQ  5
                             0ED1  1408
          00000119'EF    D6  0ED1  1409          INCL    QUEUE_COUNT             ; INCREMENT COUNT
          58   52   08   C3  0ED7  1410          SUBL3   #<LKB$L_SQFL-LKB$L_LKID>,R2,R8   ; GET START OF RELEVANT DATA
                             0EDB  1411          REQMEM  (R8),LKB_BFR,#6         ; SAVE ID, GRANT & REQUEST MODE
      57  00000008'EF    9E  0EEC  1412          MOVAB   LKB_BFR,R7
            6443     67  D0  0EF3  1413          MOVL    LKID_OFF(R7),(R4)[R3]   ; MOVE INTO STORAGE BUFFERS
          6543    05 A7  90  0EF7  1414          MOVB    GRMD_OFF(R7),(R5)[R3]
```

```
           6643   04 A7   90  0EFC   1415              MOVB    RQMD_OFF(R7),(R6)[R3]
                              0F01   1416
             BB 53   03   F2  0F01   1417              AOBLSS  #3,R3,20$                 ; IF WE'VE SAVED 3 ITEMS, PRINT THEM
       00000F2B'EF   00   FB  0F05   1418              CALLS   #0,PRINT_LINE
                     B0   11  0F0C   1419              BRB     10$                       ; GET MORE ITEMS
                              0F0E   1420
       00000F2B'EF   00   FB  0F0E   1421 30$:         CALLS   #0,PRINT_LINE             ; PRINT ANY REMAINING ITEMS
          00000119'EF     D5  0F15   1422              TSTL    QUEUE_COUNT               ; WERE ANY ITEMS PROCESSED?
                     0D   12  0F1B   1423              BNEQ    EXIT                      ; IF NEQ, YES
                              0F1D   1424              PRINT   0,<      *** EMPTY QUEUE ***>
                              0F2A   1425
                          04  0F2A   1426 EXIT:        RET
```

```
                            0F2B                .SBTTL  PRINT_LINE - OUTPUT QUEUE DATA
                            0F2B        ;+++
                            0F2B        ;
                            0F2B        ;               PRINT_LINE
                            0F2B        ;
                            0F2B        ;       PURPOSE
                            0F2B        ;               GIVEN LOCK DATA FOR ONE LINE (0 - 3 ITEMS), FORMAT AND
                            0F2B        ;               DISPLAY THIS DATA
                            0F2B        ;
                            0F2B        ;       INPUT
                            0F2B        ;               R3 - COUNT OF ITEMS TO DISPLAY
                            0F2B        ;               R4 - ADDRESS OF BUFFER CONTAINING LOCK IDS
                            0F2B        ;               R5 - ADDRESS OF BUFFER CONTAINING GRANT-MODES
                            0F2B        ;               R6 - ADDRESS OF BUFFER CONTAINING REQUESTED-MODES
                            0F2B        ;               QUEUE_TYPE - TYPE OF QUEUE WE ARE WORKING ON
                            0F2B        ;
                            0F2B        ;       OUTPUT
                            0F2B        ;               ONE DISPLAYED LINE OF QUEUE LOCK DATA.
                            0F2B        ;
                            0F2B        ;---
                            0F2B
                      0180  0F2B                .ENTRY  PRINT_LINE,^M<R7,R8>
                            0F2D
                            0F2D        ;
                            0F2D        ;       INITIALIZE
                            0F2D        ;
 57   00000198'EF     D0    0F2D                MOVL    QUEUE_TYPE,R7                   ; SAVE QUEUE TYPE
00000194'EF    53     D0    0F34                MOVL    R3,COUNT                        ; SAVE COUNT
            01 53     F4    0F3B                SOBGEQ  R3,10$                          ; CONVERT TO ARRAY INDEX
                      04    0F3E                RET                                     ; IF COUNT 0, NO ITEMS TO DISPLAY
                            0F3F
                            0F3F        ;
                            0F3F        ;       FORMAT DATA
                            0F3F        ;
               57     D5    0F3F        10$:    TSTL    R7                              ; ONLY FORMAT RQ MODE IF WAIT OR CONVERT
               09     19    0F41                BLSS    15$                             ;       QUEUE
       58    6643     9A    0F43                MOVZBL  (R6)[R3],R8
       F285 CF48      DD    0F47                PUSHL   LKMODE_TBL[R8]
                            0F4C
               57     D5    0F4C        15$:    TSTL    R7                              ; ONLY FORMAT GR MODE IF GRANT OR CONVERT
               09     14    0F4E                BGTR    20$                             ;       QUEUE
       58    6543     9A    0F50                MOVZBL  (R5)[R3],R8
       F278 CF48      DD    0F54                PUSHL   LKMODE_TBL[R8]
                            0F59
             6443     DD    0F59        20$:    PUSHL   (R4)[R3]                        ; PROCESS LOCK ID
            E0 53     F4    0F5C                SOBGEQ  R3,10$                          ; IF NOT YET DONE, LOOP
                            0F5F
    58    F0DB CF     9E    0F5F                MOVAB   CONV_STR_TBL,R8                 ; GET ADDRESS OF CORRECT FAO STRING TABLE
               57     D5    0F64                TSTL    R7
               05     13    0F66                BEQL    25$                             ; IF EQL, CONVERT
    58    F0C2 CF     9E    0F68                MOVAB   LOCK_STR_TBL,R8                 ; ELSE WAIT/GRANT
                            0F6D
 53   00000194'EF     D0    0F6D        25$:    MOVL    COUNT,R3                        ; PICK UP CORRECT STRING FOR NUMBER OF
       58    6843     D0    0F74                MOVL    (R8)[R3],R8                     ;       ITEMS
          53   53     C0    0F78                ADDL    R3,R3                           ; CALCULATE CORRECT FAO ARG COUNT
               57     D5    0F7B                TSTL    R7
               07     12    0F7D                BNEQ    30$
```

```
     53  00000194'EF   C0  0F7F          ADDL    COUNT,R3
                           0F86
                           0F86  30$:    PRINTD  r3,(r8)                ; DISPLAY LINE
                           0F91
                       04  0F91          RET
                           0F92
                           0F92          .END
```

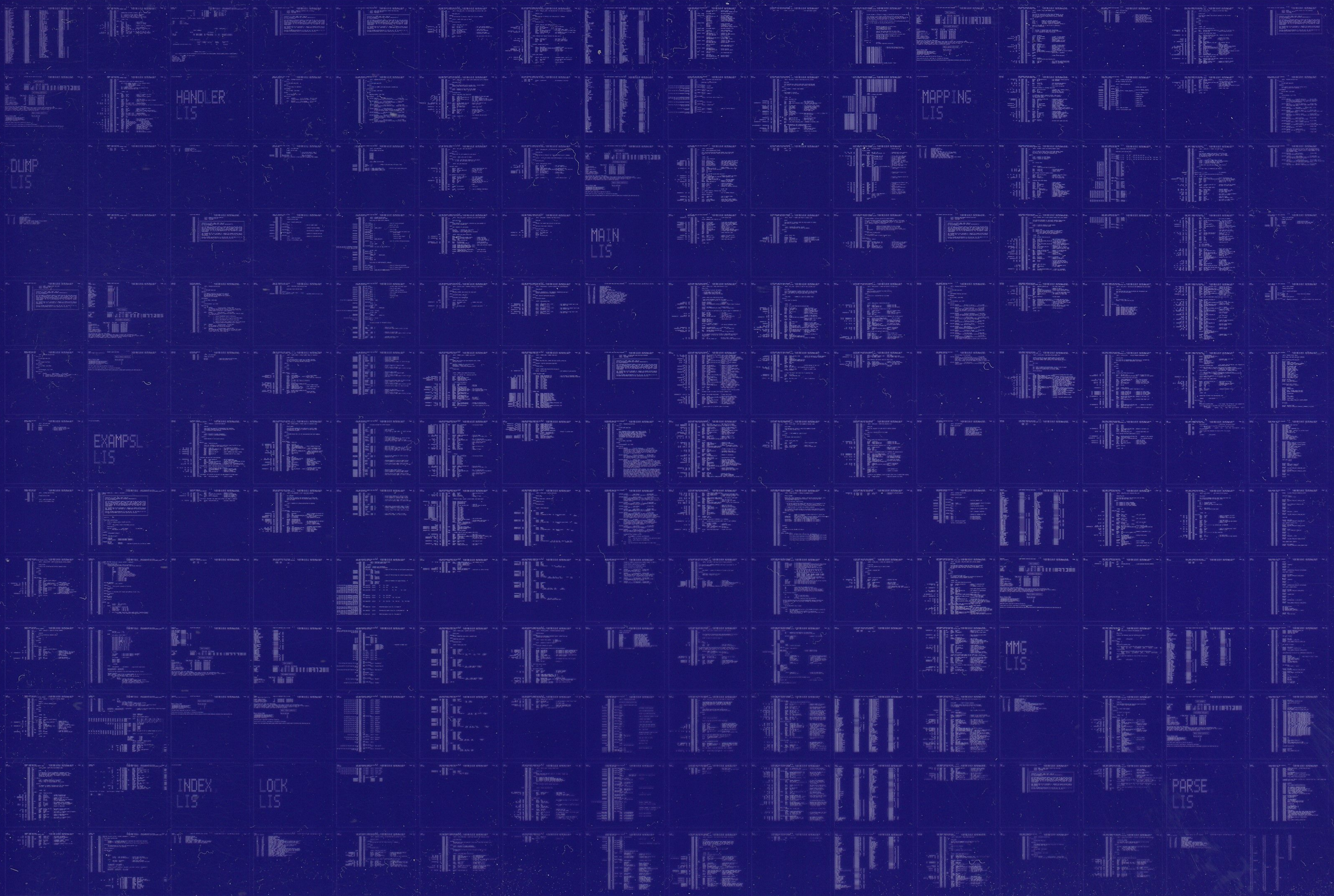
```
$$T2                        = 00000004                GROUP_TXT                   000001E0 R       02
ACMODE                        000002FF R       02     GR_STRING                   000001FB R       03
ACMODE_TBL                    00000318 R       03     HTBL_CNT                    00000190 R       02
ARGS                        = 00000003                HTBL_INDX                   0000018C R       02
BAST                          000001FF R       02     KMODE                       000002EB R       03
BLANKS                        00000220 R       03     LCK$GL_HASHTBL              ********    X  03
BLKAST                        000002BB R       02     LCK$GL_HTBLCNT              ********    X  03
BL_STRING                     00000223 R       03     LCK$GL_IDTBL                ********    X  03
BUF1                          00000227 R       02     LCK$GL_MAXID                ********    X  03
BUF2                          0000023B R       02     LIB$SIGNAL                  ********    X  03
BUF3                          00000253 R       02     LINE_COUNT                  ********    X  03
BUF4                          0000026B R       02     LKB                         000002BF R       02
CGMOD                         000001F7 R       02     LKB$B_GRMODE              = 00000035
CHF$L_MCH_SAVR0             = 0000000C                LKB$B_RQMODE              = 00000034
CHF$L_SIG_NAME              = 00000004                LKB$B_STATE               = 00000036
CONVERT                     = 00000000                LKB$K_LENGTH              = 00000060
CONVSTR                       0000016B R       03     LKB$L_BLKASTADR           = 00000020
CONVSTR1                      000000C6 R       03     LKB$L_CSID                = 00000058
CONVSTR2                      000000DF R       03     LKB$L_LKID                = 00000030
CONVSTR3                      00000108 R       03     LKB$L_OWNQFL              = 00000040
CONV_STR_TBL                  0000003E R       03     LKB$L_PARENT              = 00000048
COUNT                         00000194 R       02     LKB$L_PID                 = 0000000C
CREAD                         000001EC R       03     LKB$L_REMLKID             = 00000054
CSID                          0000023F R       02     LKB$L_RSB                 = 00000050
CV_STRING                     00000206 R       03     LKB$L_SQFL                = 00000038
CWRITE                        000001EF R       03     LKB$V_MSTCPY              = 00000004
DIRENTRY                      00000257 R       02     LKB$W_FLAGS               = 00000028
DIR_ENTRY                     000002DB R       03     LKB$W_REFCNT              = 0000004C
DISPLAY_LOCK                  0000051D RG      03     LKB$W_STATUS              = 0000002A
DMP1                          000002CF R       02     LKB_BFR                     00000008 R       02
DMP3                          000002EF R       02     LKB_RSB_BFR                 0000006C R       02
DMP5                          0000030F R       02     LKID                        0000026F R       02
DMP7                          0000032F R       02     LKID_BFR                    000001A2 R       02
DONE                          00000442 R       03     LKID_OFF                  = 00000000
EMODE                         000002F7 R       03     LKMODE_TBL                  000001D1 R       03
EX                            000001F8 R       03     LOCAL                       000002BA R       03
EXIT                          00000F2A R       03     LOCKID                      00000090 RG      02
FAO_GROUP_DSC                 000001AE R       02     LOCKMODE_TBL                000001C5 R       03
FAO_RMINFO_DSC                00000094 R       02     LOCKSTR1                    0000005A R       03
FLAGS1                        00000277 R       02     LOCKSTR2                    0000006F R       03
FLAGS2                        00000293 R       02     LOCKSTR3                    00000093 R       03
FLAGS3                        000002AF R       02     LOCK_COND_HAND              000009E8 RG      03
FLAGS_TBL                     0000022A R       03     LOCK_COUNT                  00000000 R       02
FORMAT_LOCK                   00000556 RG      03     LOCK_HEAD                   00000000 R       03
FORMAT_RSB                    00000BCC RG      03     LOCK_STR_TBL                0000002E R       03
GETLCK                        000003C7 R       03     MASTER                      000002C8 R       03
GET_LKB                       00000443 RG      03     MODE1                       0000028F R       02
GGMOD                         000001EF R       02     MODE2                       000002AB R       02
GRANT                       = FFFFFFFF                MSG$_LOCKIDZER              ********    X  03
GRANTSTR                      00000141 R       03     MSG$_NOLKB                  ********    X  03
GRMD_BFR                      0000019C R       02     MSG$_NOLOCKS                ********    X  03
GRMD_OFF                    = 00000005                MSG$_NOPRLOCK               ********    X  03
GROUP_BFR                     0000006C R       02     MSG$_NORESOURC              ********    X  03
GROUP_BUF                     000001BA R       02     MSG$_OUTOFRANG              ********    X  03
GROUP_BUF_DSC                 000001C0 R       02     MSG$_SUCCESS                ********    X  03
GROUP_DSC                     000001D8 R       02     NEW_PAGE                    ********    X  03
GROUP_NUM                     000001E8 R       02     NOLCK                       00000430 R       03
```

```
NOT_VALID                         000002D1 R       03          RSB$B_RMOD                        = 0000004E
NSPACE                            0000031F R       02          RSB$B_RSNLEN                      = 0000004F
NULL                              000001E9 R       03          RSB$K_LENGTH                      = 00000050
NULL_CSTRING                      00000222 R       03          RSB$K_MAXLEN                      = 0000001F
PAGE_SIZE                         ********     X   03          RSB$L_CSID                        = 00000038
PARID                             00000283 R       02          RSB$L_CVTQFL                      = 00000018
PARID_BFR                         00000004 R       02          RSB$L_GRQFL                       = 00000010
PART2                             0000067E R       03          RSB$L_PARENT                      = 00000048
PCB$L_LOCKQFL                   = 00000104                     RSB$L_VALSEQNUM                   = 0000003C
PID                               00000273 R       02          RSB$L_WTQFL                       = 00000020
PREAD                             000001F2 R       03          RSB$Q_VALBLK                      = 00000028
PRINT                             ********     X   03          RSB$T_RESNAM                      = 00000050
PRINT_LINE                        00000F2B RG      03          RSB$V_DIRENTRY                    = 00000000
PRINT_LOCK                        00000856 RG      03          RSB$V_VALINVLD                    = 00000001
PRINT_RSB                         00000D34 RG      03          RSB$W_BLKASTCNT                   = 00000042
PROCESS                           000002C0 R       03          RSB$W_GROUP                       = 0000004C
PROCESS_QUEUE                     00000E55 RG      03          RSB$W_REFCNT                      = 00000040
PRSB                              000001F3 R       02          RSB$W_STATUS                      = 0000000E
PWRITE                            000001F5 R       03          RSBCSID_BFR                         00000068 R       02
QUEUE_COUNT                       00000119 R       02          RSB_BFR                             0000011D R       02
QUEUE_TYPE                        00000198 R       02          RSNLEN_BFR                          0000006F R       02
QUE_STR_TBL                       0000004E R       03          SAVE_LOCK_DATA                      000004B7 RG      03
RACMOD                            00000243 R       02          SEQNUM                              00000213 R       02
REQMEM                            ********     X   03          SET_HEADING                         ********     X   03
RES1                              000002C7 R       02          SHOW_ALL_LOCKS                      00000328 RG      03
RES2                              000002C3 R       02          SHOW_ALL_RES                        00000A32 RG      03
RES3                              000002E7 R       02          SHOW_ONE_LOCK                       0000036E RG      03
RES4                              000002E3 R       02          SHOW_ONE_RES                        00000AF3 RG      03
RES5                              00000307 R       02          SHOW_PROC_LOCK                      000003AE RG      03
RES6                              00000303 R       02          SHOW_QUEUES                         00000B92 RG      03
RES7                              00000327 R       02          SHOW_RSB                            00000B62 RG      03
RES8                              00000323 R       02          SKIP_LINES                          ********     X   03
RESN1                             0000021F R       02          SMODE                               00000302 R       03
RESN2                             0000021B R       02          SPACE                               000002CF R       03
RESN3                             00000233 R       02          SRSBCT                              000001FB R       02
RESN4                             0000022F R       02          SS$_RESIGNAL                        ********     X   03
RESN5                             0000024B R       02          SS$_UNWIND                          ********     X   03
RESN6                             00000247 R       02          STATE1                              00000287 R       02
RESN7                             00000263 R       02          STATE2                              000002A3 R       02
RESN8                             0000025F R       02          STATUS_TBL                          00000272 R       03
RESNAM_BFR                        00000070 R       02          STS1                                000002D3 R       02
RESNLEN                           0000022B R       02          STS2                                000002F3 R       02
RES_COUNT                         00000115 R       02          STS3                                00000313 R       02
RES_HEAD                          00000015 R       03          SUBLKS                              0000029F R       02
RLEN                              000002DF R       02          SYS$FAO                             ********     X   03
RMINFO                            00000337 R       02          SYS$UNWIND                          ********     X   03
RMINFO_BFR                        000000C5 R       02          SYS_DSC                             000001C8 R       02
RMINFO_CNT                        000000C4 R       02          TYPE                                00000333 R       02
RMINFO_DSC                        000000BC R       02          UMODE                               0000030E R       03
RMINFO_LEN                        000000B8 R       02          VAL1                                0000020F R       02
RMOD_BFR                          0000006E R       02          VAL2                                0000020B R       02
RNSPACE                           0000025B R       02          VAL3                                00000207 R       02
RQMD_BFR                          0000019F R       02          VAL4                                00000203 R       02
RQMD_OFF                        = 00000004                     VALID                               00000217 R       02
RSB                               000001EB R       02          WAIT                              = 00000001
RSB$B_CGMODE                    = 0000000D                     WAITSTR                             0000019B R       03
RSB$B_GGMODE                    = 0000000C                     WT_STRING                           00000214 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
SDADATA                             0000033B  (    827.)  02 (  2.)  NOPIC    USR   CON   REL   LCL NOSHR NOEXE   RD     WRT NOVEC LONG
LOCK                                00000F92  (   3986.)  03 (  3.)  NOPIC    USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC LONG
LITERALS                            0000034C  (    844.)  04 (  4.)  NOPIC    USR   CON   REL   LCL NOSHR   EXE   RD   NOWRT NOVEC BYTE

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

Phase                         Page faults    CPU Time       Elapsed Time
-----                         -----------    --------       ------------
Initialization                         29    00:00:00.05    00:00:01.24
Command processing                    107    00:00:00.45    00:00:03.92
Pass 1                                310    00:00:06.76    00:00:26.79
Symbol table sort                       0    00:00:00.69    00:00:01.98
Pass 2                                264    00:00:02.72    00:00:10.19
Symbol table output                    26    00:00:00.14    00:00:00.30
Psect synopsis output                   3    00:00:00.02    00:00:00.02
Cross-reference output                  0    00:00:00.00    00:00:00.00
Assembler run totals                  741    00:00:10.83    00:00:44.44

The working set limit was 1650 pages.
66653 bytes (131 pages) of virtual memory were used to buffer the intermediate code.
There were 40 pages of symbol table space allocated to hold 660 non-local and 73 local symbols.
1491 source lines were read in Pass 1, producing 79 object records in Pass 2.
26 pages of virtual memory were used to define 24 macros.

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

Macro library name                                 Macros defined
------------------                                 --------------
_$255$DUA28:[SDA.OBJ]SDALIB.MLB;1                               9
_$255$DUA28:[SYS.OBJ]LIB.MLB;1                                  3
_$255$DUA28:[SYSLIB]STARLET.MLB;2                               9
TOTALS (all libraries)                                         21

660 GETS were required to define 21 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS$:LOCK/OBJ=OBJ$:LOCK MSRC$:LOCK/UPDATE=(ENH$:LOCK)+EXECML$/LIB+LIB$:SDALIB/LIB
```

0352 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY



HANDLER
LIS

MAPPING
LIS

DUMP
LIS

MAIN
LIS

EXAMPSL
LIS

MMG
LIS

INDEX
LIS

LOCK
LIS

PARSE
LIS