
MMM MMM TTTTTTTTTTTTTTT AAAAAAAAA AAAAAAAAA CCCCCCCCCCCC PPPPPPPPPPPP
MMM MMM TTTTTTTTTTTTTTT AAAAAAAAA AAAAAAAAA CCCCCCCCCCCC PPPPPPPPPPPP
MMM MMM TTTTTTTTTTTTTTT AAAAAAAAA AAAAAAAAA CCCCCCCCCCCC PPPPPPPPPPPP
MMMMMM MMMMMM TTT AAA AAA AAA AAA CCC PPP PPP
MMMMMM MMMMMM TTT AAA AAA AAA AAA CCC PPP PPP
MMMMMM MMMMMM TTT AAA AAA AAA AAA CCC PPP PPP
MMM MMM MMM TTT AAA AAA AAA AAA CCC PPP PPP
MMM MMM MMM TTT AAA AAA AAA AAA CCC PPP PPP
MMM MMM MMM TTT AAA AAA AAA AAA CCC PPP PPP
MMM MMM TTT AAA AAA AAA AAA CCC PPPPPPPPPPPP
MMM MMM TTT AAA AAA AAA AAA CCC PPPPPPPPPPPP
MMM MMM TTT AAA AAA AAA AAA CCC PPPPPPPPPPPP
MMM MMM TTT AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA CCC PPP
MMM MMM TTT AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA CCC PPP
MMM MMM TTT AAAAAAAAAAAAAAA AAAAAAAAAAAAAAA CCC PPP
MMM MMM TTT AAA AAA AAA AAA CCC PPP
MMM MMM TTT AAA AAA AAA AAA CCC PPP
MMM MMM TTT AAA AAA AAA AAA CCC PPP
MMM MMM TTT AAA AAA AAA AAA CCCCCCCCCCCC PPP
MMM MMM TTT AAA AAA AAA AAA CCCCCCCCCCCC PPP
MMM MMM TTT AAA AAA AAA AAA CCCCCCCCCCCC PPP

```
  CCCCCCCC  HH      HH  KK     KK  DDDDDDDD    MM      MM    OOOOOO
  CCCCCCCC  HH      HH  KK     KK  DDDDDDDD    MM      MM    OOOOOO
CC          HH      HH  KK     KK  DD      DD  MMMM  MMMM  OO      OO
CC          HH      HH  KK     KK  DD      DD  MMMM  MMMM  OO      OO
CC          HH      HH  KK   KK    DD      DD  MM  MM  MM  OO      OO
CC          HH      HH  KK   KK    DD      DD  MM  MM  MM  OO      OO
CC          HHHHHHHHHH  KKKKK      DD      DD  MM      MM  OO      OO
CC          HHHHHHHHHH  KKKKK      DD      DD  MM      MM  OO      OO
CC          HH      HH  KK   KK    DD      DD  MM      MM  OO      OO
CC          HH      HH  KK   KK    DD      DD  MM      MM  OO      OO
CC          HH      HH  KK     KK  DD      DD  MM      MM  OO      OO
CC          HH      HH  KK     KK  DD      DD  MM      MM  OO      OO
  CCCCCCCC  HH      HH  KK     KK  DDDDDDDD    MM      MM    OOOOOO
  CCCCCCCC  HH      HH  KK     KK  DDDDDDDD    MM      MM    OOOOOO

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
```

```
;     1          0001  0 MODULE CHKDMO (LANGUAGE (BLISS32) ,
;     2          0002  0                  IDENT = 'V04-000'
;     3          0003  0                  ) =
;     4          0004  1 BEGIN
;     5          0005  1 !
;     6          0006  1 !**************************************************************************
;     7          0007  1 !*                                                                        *
;     8          0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                              *
;     9          0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                *
;    10          0010  1 !*  ALL RIGHTS RESERVED.                                                  *
;    11          0011  1 !*                                                                        *
;    12          0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED  *
;    13          0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    14          0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    15          0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    16          0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    17          0017  1 !*  TRANSFERRED.                                                           *
;    18          0018  1 !*                                                                        *
;    19          0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    20          0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    21          0021  1 !*  CORPORATION.                                                           *
;    22          0022  1 !*                                                                        *
;    23          0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
;    24          0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;    25          0025  1 !*                                                                        *
;    26          0026  1 !*                                                                        *
;    27          0027  1 !**************************************************************************
;    28          0028  1
;    29          0029  1 !++
;    30          0030  1 !
;    31          0031  1 ! FACILITY:  VAS/VMS MTAACP
;    32          0032  1 !
;    33          0033  1 ! ABSTRACT:
;    34          0034  1 !
;    35          0035  1 !       This routine dismounts the volume in use if it should be.
;    36          0036  1 !
;    37          0037  1 ! ENVIRONMENT:
;    38          0038  1 !
;    39          0039  1 !       VAX/VMS operating system, including privileged system services
;    40          0040  1 !       and internal exec routines.
;    41          0041  1 !
;    42          0042  1 !--
;    43          0043  1
;    44          0044  1 !
;    45          0045  1 !
;    46          0046  1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  29-Apr-1977  17:19
;    47          0047  1 !
;    48          0048  1 ! MODIFIED BY:
;    49          0049  1 !
;    50          0050  1 !       V03-010 HH0049          Hai Huang                   16-Aug-1984
;    51          0051  1 !               Call IOC$DALLOC_DMT to handle deallocation on dismount.
;    52          0052  1 !
;    53          0053  1 !       V03-009 MMD0324         Meg Dumont,     15-Aug-1984  14:37
;    54          0054  1 !               Fix to index off the UCBLIST with the NVOL rather than
;    55          0055  1 !               the volume number from the MVL. The MVL may not be valid
;    56          0056  1 !               thus we shouldn't be using it.
;    57          0057  1 !
```

```
;    58          0058  1 !        V03-008 ACG0441         Andrew C. Goldstein,    13-Aug-1984  12:12
;    59          0059  1 !                Issue IO$_AVAILABLE function after IO$_UNLOAD to release
;    60          0060  1 !                drive correctly.
;    61          0061  1 !
;    62          0062  1 !        V03-007 ACG0441         Andrew C. Goldstein,    11-Aug-1984  16:58
;    63          0063  1 !                Rework dismount interlocking to eliminate races and
;    64          0064  1 !                uninterlocked procesing. Clear device lock value block on
;    65          0065  1 !                dismount.
;    66          0066  1 !
;    67          0067  1 !        V03-006 MMD0289         Meg Dumont,     10-Apr-1984  14:15
;    68          0068  1 !                Fix to use IO$_AVAILABLE instead of setting to SYNC
;    69          0069  1 !                IPL and clearing the UCB VALID bit. Also fixed the
;    70          0070  1 !                DALLOC_DEV linkage to indicate that R3 is trashed across
;    71          0071  1 !                the call. Removed any knowledge inside this routine
;    72          0072  1 !                of setting to IPL$_SYNC.
;    73          0073  1 !
;    74          0074  1 !        V03-005 LMP0221         L. Mark Pilant,         28-Mar-1984  13:04
;    75          0075  1 !                Change UCB$L_OWNUIC to ORB$L_OWNER and UCB$W_VPROT to
;    76          0076  1 !                ORB$W_PROT.
;    77          0077  1 !
;    78          0078  1 !        V03-004 MMD0281         Meg Dumont,     23-Mar-1984  10:29
;    79          0079  1 !                Fix to pass address of UCB on IOC$DALLC_DEV call.
;    80          0080  1 !
;    81          0081  1 !        V03-003 ACG0399         Andrew C. Goldstein,    27-Feb-1984  13:24
;    82          0082  1 !                Rename EXE$UNLOCK_DEV to IOC$UNLOCK_DEV
;    83          0083  1 !
;    84          0084  1 !        V03-002 TCM0001         Trudy C. Matthews       10-May-1983
;    85          0085  1 !                Call routine EXE$DALLOC_DEV to correctly do device
;    86          0086  1 !                deallocation in a cluster.
;    87          0087  1 !
;    88          0088  1 !        V03-001 STJ0263         Steven T. Jeffreys      22-Apr-1982
;    89          0089  1 !                Do not mung the device allocation access mode.
;    90          0090  1 !
;    91          0091  1 !        V02-010 DMW00035        David Michael Walp      15-Sep-1981
;    92          0092  1 !                Fixed Cancel I/O vs Dismount race condition
;    93          0093  1 !
;    94          0094  1 !        V02-009 DMW00026        David Michael Walp      20-Jul-1981
;    95          0095  1 !                Changes to RET_FREE_PAGE, new parameter to say
;    96          0096  1 !                that the P0 space will should be contracted.
;    97          0097  1 !
;    98          0098  1 !        V02-008 DMW00011        David Michael Walp      14-Mar-1981
;    99          0099  1 !                Changed calculation of CCB address to GET_CCB
;   100          0100  1 !
;   101          0101  1 !        V02-007 REFORMAT        Maria del C. Nasr       30-Jun-1980
;   102          0102  1 !
;   103          0103  1 !**
;   104          0104  1
;   105          0105  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;   106          0106  1
;   107          0107  1 REQUIRE 'SRC$:MTADEF.B32';
;   108          0491  1
;   109          0492  1 !
;   110          0493  1 ! Part of this routine use to run at IPL$_SYNCH, however we will still lock
;   111          0494  1 ! the code down.
;   112          0495  1 !
;   113          0496  1 LOCK_CODE;
;   114          0497  1
```

```
;  116     0498  1 GLOBAL ROUTINE CHECK_DISMOUNT (UCB) : COMMON_CALL NOVALUE =
;  117     0499  1
;  118     0500  1 !++
;  119     0501  1 !
;  120     0502  1 ! FUNCTIONAL DESCRIPTION:
;  121     0503  1 !
;  122     0504  1 !       This routine checks if the volume in use is marked for dismount and
;  123     0505  1 !       idle. If so, it completes the dismount.
;  124     0506  1 !
;  125     0507  1 ! CALLING SEQUENCE:
;  126     0508  1 !       CHECK_DISMOUNT (ARG1)
;  127     0509  1 !
;  128     0510  1 ! INPUT PARAMETERS:
;  129     0511  1 !       ARG1 - address of unit control block for primary UCB
;  130     0512  1 !
;  131     0513  1 ! IMPLICIT INPUTS:
;  132     0514  1 !       QUEUE_HEAD: queue header for ACP
;  133     0515  1 !
;  134     0516  1 ! OUTPUT PARAMETERS:
;  135     0517  1 !       None
;  136     0518  1 !
;  137     0519  1 ! IMPLICIT OUTPUTS:
;  138     0520  1 !       None
;  139     0521  1 !
;  140     0522  1 ! ROUTINE VALUE:
;  141     0523  1 !       None
;  142     0524  1 !
;  143     0525  1 ! SIDE EFFECTS:
;  144     0526  1 !       Volume dismounted if appropriate
;  145     0527  1 !
;  146     0528  1 !--
;  147     0529  1
;  148     0530  2     BEGIN
;  149     0531  2
;  150     0532  2     BUILTIN
;  151     0533  2         TESTBITSC;
;  152     0534  2
;  153     0535  2     LINKAGE
;  154     0536  2         DALLOC_DEV      = JSB (REGISTER=4,REGISTER=5;) :
;  155     0537  2                           NOPRESERVE (3)
;  156     0538  2                           PRESERVE (2,4,5)
;  157     0539  2                           NOTUSED (6,7,8,9,10,11);
;  158     0540  2
;  159     0541  2     EXTERNAL REGISTER
;  160     0542  2         COMMON_REG;
;  161     0543  2
;  162     0544  2     MAP
;  163     0545  2         UCB     : REF BBLOCK;           ! address of unit control block
;  164     0546  2
;  165     0547  2     LOCAL
;  166     0548  2         CCB     : REF BBLOCK,           ! address of channel control block
;  167     0549  2         STS,                            ! general status value
;  168     0550  2         LKSTS           : VECTOR [6],   ! lock status block
;  169     0551  2                                         ! address of mag tape volume entry
;  170     0552  2         UNLOAD,                         ! flag to indicate unloading
;  171     0553  2         PAGE,
;  172     0554  2         ORB     : REF BBLOCK,           ! ORB address
```

```
;  173     0555  2          UCBLIST : REF VECTOR,   ! vector of ucb's allocated to volume set
;  174     0556  2          VCB     : REF BBLOCK;                  ! local address of VCB
;  175     0557  2
;  176     0558  2      EXTERNAL
;  177     0559  2
;  178     0560  2          IO_CHANNEL,                            ! assign channel for tape I/O
;  179     0561  2          SCH$GL_CURPCB   : REF BBLOCK ADDRESSING_MODE (GENERAL),
;  180     0562  2          QUEUE_HEAD      : REF BBLOCK;          ! address of ACP queue header
;  181     0563  2
;  182     0564  2      EXTERNAL ROUTINE
;  183     0565  2
;  184     0566  2          IOC$DALLOC_DMT  : DALLOC_DEV ADDRESSING_MODE (GENERAL),
;  185     0567  2                                    ! deallocate device
;  186     0568  2          RET_FREE_PAGE   : COMMON_CALL,  ! return free page to virtual mem pool
;  187     0569  2          LOCK_IODB,                      ! lock I/O data base mutex
;  188     0570  2          SEND_ERRLOG,
;  189     0571  2          UNLOCK_IODB,                    ! unlock I/O data base mutex
;  190     0572  2          DEALLOCATE,                     ! deallocate dynamic memory
;  191     0573  2          GET_CCB;                        ! get address of the CCB
;  192     0574  2
;  193     0575  2
;  194     0576  2      ! First check if the volume is marked for dismount
;  195     0577  2      !
;  196     0578  2
;  197     0579  2      IF NOT .BBLOCK[UCB[UCB$L_DEVCHAR], DEV$V_DMT]
;  198     0580  2      THEN
;  199     0581  2          RETURN;
;  200     0582  2
;  201     0583  2      VCB = .UCB[UCB$L_VCB];                     ! pickup VCB address
;  202     0584  2
;  203     0585  2      ! The volume is marked for dismount. The remainder of the tests and the
;  204     0586  2      ! dismount bit twiddling must be done interlocked.
;  205     0587  2      !
;  206     0588  2      LOCK_IODB();
;  207     0589  2      SET_IPL (IPL$_SYNCH);
;  208     0590  2
;  209     0591  2      ! If a cancel I/O is pending we can not dismount the volume, the volume
;  210     0592  2      ! will be check for dismount when the cancel I/O is completed.  This stops
;  211     0593  2      ! the crash caused by cancel and dismount happening while the ACP has a QIO
;  212     0594  2      ! with an completion AST outstanding to the volume.
;  213     0595  2      !
;  214     0596  3      IF (.VCB[VCB$W_TRANS] EQL 1) AND (NOT .VCB[VCB$V_CANCELIO])
;  215     0597  2      THEN
;  216     0598  3          BEGIN
;  217     0599  3
;  218     0600  3          ! The volume is marked for dismount and idle.  Mark all UCB's with
;  219     0601  3          ! dismount in progress to stop all further activity.
;  220     0602  3          !
;  221     0603  3          UCBLIST = BBLOCK[.VCB[VCB$L_RVT], RVT$L_UCBLST];
;  222     0604  3
;  223     0605  3          DECR NVOL FROM .BBLOCK[.VCB[VCB$L_RVT], RVT$B_NVOLS] - 1 TO 0 DO
;  224     0606  4              BEGIN
;  225     0607  4              UCB = .UCBLIST[.NVOL];             ! UCB from RVT list
;  226     0608  4              UCB[UCB$V_DISMOUNT] = 1;
;  227     0609  3              END;
;  228     0610  3
;  229     0611  3          UNLOCK_IODB ();
```

```
;  230          0612  3
;  231          0613  3          ! Establish whether volumes are to be unloaded. The primary UCB
;  232          0614  3          ! contains the unload flag.
;  233          0615  3          !
;  234          0616  3          IF TESTBITSC (BBLOCK [.UCBLIST[0], UCB$V_UNLOAD])
;  235          0617  3          THEN UNLOAD = 1
;  236          0618  3          ELSE UNLOAD = 0;
;  237          0619  3
;  238          0620  3          ! Loop through the UCB's again. For each one, send the dismount
;  239          0621  3          ! error log message and then unload the unit if it is online.
;  240          0622  3          !
;  241          0623  3          CCB = GET_CCB (.IO_CHANNEL);
;  242          0624  3          DECR NVOL FROM .BBLOCK[.VCB[VCB$L_RVT], RVT$B_NVOLS] - 1 TO 0 DO
;  243          0625  4              BEGIN
;  244          0626  4              UCB = .UCBLIST[.NVOL];                  ! UCB from RVT list
;  245          0627  4              SEND_ERRLOG(0, .UCB);
;  246          0628  4
;  247          0629  4              CCB[CCB$L_UCB] = .UCBLIST[.NVOL];
;  248          0630  4              IF .UNLOAD
;  249          0631  4              THEN $QIOW(CHAN = .IO_CHANNEL, FUNC = IO$_UNLOAD);
;  250          0632  4              $QIOW(CHAN = .IO_CHANNEL, FUNC = IO$_AVAILABLE);
;  251          0633  4
;  252          0634  4              ! If the UCB is not allocated, acquire the device lock so
;  253          0635  4              ! that its value block can be written.
;  254          0636  4              !
;  255          0637  4              IF (LKSTS [1] = .UCB [UCB$L_LOCKID]) NEQ 0
;  256          0638  4                  AND .UCB [UCB$L_PID] EQL 0
;  257          0639  4              THEN
;  258          0640  5                  BEGIN
;  259        P 0641  5                  STS = $ENQW (LKMODE = LCK$K_PWMODE,
;  260        P 0642  5                               LKSB   = LKSTS,
;  261        P 0643  5                               EFN    = EFN,
;  262        P 0644  5                               FLAGS  = LCK$M_CONVERT + LCK$M_SYNCSTS
;  263          0645  5                                 + LCK$M_NOQUOTA);
;  264          0646  5                  IF NOT .STS
;  265          0647  5                  OR NOT .LKSTS
;  266          0648  5                  THEN BUG_CHECK (XQPERR, FATAL, 'Unexpected lock manager error');
;  267          0649  4                  END;
;  268          0650  4
;  269          0651  4              ! Now complete the dismount.  Mark primary unit and secondary
;  270          0652  4              ! units dismounted and deallocate those units which should be
;  271          0653  4              ! on dismount.
;  272          0654  4              !
;  273          0655  4              ORB = .UCB[UCB$L_ORB];
;  274          0656  4              LOCK_IODB ();
;  275          0657  4
;  276          0658  4              BBLOCK[UCB[UCB$L_DEVCHAR], DEV$V_MNT] = 0;
;  277          0659  4              BBLOCK[UCB[UCB$L_DEVCHAR], DEV$V_DMT] = 0;
;  278          0660  4              BBLOCK[UCB[UCB$L_DEVCHAR], DEV$V_SWL] = 0;
;  279          0661  4              UCB[UCB$L_VCB] = 0;
;  280          0662  4              UCB[UCB$V_DISMOUNT] = 0;
;  281          0663  4              UCB[UCB$W_REFC] = .UCB[UCB$W_REFC] - 1;
;  282          0664  4              ORB[ORB$L_SYS_PROT] = 0;
;  283          0665  4              ORB[ORB$L_OWN_PROT] = 0;
;  284          0666  4              ORB[ORB$L_GRP_PROT] = 0;
;  285          0667  4              ORB[ORB$L_WOR_PROT] = 0;
;  286          0668  4              ORB[ORB$L_OWNER] = 0;
```

```
;  287          0669  4          ! If the device lock exists, now demote it as appropriate (to CR if
;  288          0670  4          ! the device is not allocated, to EX otherwise). Clear the value
;  289          0671  4          ! block if this is the final dismount.
;  290          0672  4          !
;  291          0673  4
;  292          0674  4          IF .LKSTS [1] NEQ 0
;  293          0675  4          THEN
;  294          0676  4              BEGIN
;  295          0677  5              LKSTS [2] = 0;
;  296          0678  5              LKSTS [3] = 0;
;  297          0679  5              LKSTS [4] = 0;
;  298          0680  5              LKSTS [5] = 0;
;  299          0681  5
;  300          0682  5              STS = $ENQ (LKMODE = IF .UCB [UCB$L_PID] NEQ 0
;  301        P 0683  5                                   THEN LCK$K_EXMODE
;  302        P 0684  5                                   ELSE LCK$K_CRMODE,
;  303        P 0685  5                          LKSB    = LKSTS,
;  304        P 0686  5                          EFN     = EFN,
;  305        P 0687  5                          FLAGS   = LCK$M_CONVERT + LCK$M_CVTSYS
;  306        P 0688  5                                  + LCK$M_SYNCSTS + LCK$M_NOQUOTA + LCK$M_VALBLK
;  307        P 0689  5                          );
;  308          0690  5              IF NOT .STS
;  309          0691  5              OR NOT .LKSTS
;  310          0692  5              THEN BUG_CHECK (XQPERR, FATAL, 'Unexpected lock manager error');
;  311          0693  5              END;
;  312          0694  4
;  313          0695  4          ! Call IOC$DALLOC_DMT routine to deallocate the device when appropriate.
;  314          0696  4          !
;  315          0697  4
;  316          0698  4              IOC$DALLOC_DMT (.SCH$GL_CURPCB, .UCB);
;  317          0699  4
;  318          0700  4              UNLOCK_IODB ();
;  319          0701  4              END;
;  320          0702  3
;  321          0703  3          ! Finally deallocate the volume set structures.
;  322          0704  3          !
;  323          0705  3          DEALLOCATE(.VCB[VCB$L_RVT]);
;  324          0706  3          DEALLOCATE(.VCB[VCB$L_MVL]);
;  325          0707  3
;  326          0708  3          ! return volume virtual page
;  327          0709  3          !
;  328          0710  3          WHILE 1
;  329          0711  3          DO
;  330          0712  3              BEGIN
;  331          0713  4              IF REMQUE(.VCB[VCB$L_VPFL], PAGE)
;  332          0714  4              THEN EXITLOOP;
;  333          0715  4              RET_FREE_PAGE(.PAGE,TRUE);
;  334          0716  4              END;
;  335          0717  3
;  336          0718  3          DEALLOCATE(.VCB);                           ! release the VCB
;  337          0719  3          QUEUE_HEAD[AQB$B_MNTCNT] = .QUEUE_HEAD[AQB$B_MNTCNT] - 1;
;  338          0720  3          END                                         ! end of dismount processing
;  339          0721  3
;  340          0722  2      ELSE
;  341          0723  2          UNLOCK_IODB();
;  342          0724  2
;  343          0725  1      END;                                    ! end of routine CHECK_DISMOUNT
```

```
                                                  .TITLE   CHKDMO
                                                  .IDENT   \V04-000\

                                                  .EXTRN   IO_CHANNEL, SCH$GL_CURPCB
                                                  .EXTRN   QUEUE_HEAD, IOC$DALLOC_DMT
                                                  .EXTRN   RET_FREE_PAGE, LOCK_IODB
                                                  .EXTRN   SEND_ERRLOG, UNLOCK_IODB
                                                  .EXTRN   DEALLOCATE, GET_CCB
                                                  .EXTRN   SYS$QIOW, SYS$ENQW
                                                  .EXTRN   BUG$_XQPERR, SYS$ENQ

                                                  .PSECT   $LOCKEDC1$,NOWRT,2

                          07FC 00000              .ENTRY   CHECK_DISMOUNT, Save R2,R3,R4,R5,R6,R7,R8,-  : 0498
                                                           R9,R10                                       :
                 5E        1C  C2 00002           SUBL2    #28, SP                                      :
                 50   04   AC  D0 00005           MOVL     UCB, R0                                      : 0579
01         3A    A0        05  E0 00009           BBS      #5, 58(R0), 1$                               :
                               04 0000E           RET                                                   :
                 56   34   A0  D0 0000F 1$:       MOVL     52(R0), VCB                                  : 0583
        0000G    CF        00  FB 00013           CALLS    #0, LOCK_IODB                                : 0588
                 12        08  DA 00018           MTPR     #8, #18                                      : 0589
                 01   0C   A6  B1 0001B           CMPW     12(VCB), #1                                  : 0596
                           03  13 0001F           BEQL     3$                                           :
                         0175  31 00021 2$:       BRW      21$                                          :
F8         0B    A6        05  E0 00024 3$:       BBS      #5, 11(VCB), 2$                              :
                 53   20   A6  D0 00029           MOVL     32(VCB), R3                                  : 0603
                 52   44   A3  9E 0002D           MOVAB    68(R3), UCBLIST                              :
                 50   0B   A3  9A 00031           MOVZBL   11(R3), NVOL                                 : 0607
                           0D  11 00035           BRB      5$                                           :
           04    AC      6240  D0 00037 4$:       MOVL     (UCBLIST)[NVOL], UCB                         :
                 51   04   AC  D0 0003C           MOVL     UCB, R1                                      : 0608
           66    A1        10  88 00040           BISB2    #16, 102(R1)                                 :
                 F0        50  F4 00044 5$:       SOBGEQ   NVOL, 4$                                     : 0605
        0000G    CF        00  FB 00047           CALLS    #0, UNLOCK_IODB                              : 0611
                 50        62  D0 0004C           MOVL     (UCBLIST), R0                                : 0616
05         64    A0        0C  E5 0004F           BBCC     #12, 100(R0), 6$                             :
                 59        01  D0 00054           MOVL     #1, UNLOAD                                   : 0617
                           02  11 00057           BRB      7$                                           :
                           59  D4 00059 6$:       CLRL     UNLOAD                                       : 0618
                  0000G    CF  DD 0005B 7$:       PUSHL    IO_CHANNEL                                   : 0623
        0000G    CF        01  FB 0005F           CALLS    #1, GET_CCB                                  :
                 6E        50  D0 00064           MOVL     R0, CCB                                      :
                 58   0B   A3  9A 00067           MOVZBL   11(R3), NVOL                                 : 0624
                         00F2  31 0006B           BRW      16$                                          :
           04    AC      6248  D0 0006E 8$:       MOVL     (UCBLIST)[NVOL], UCB                         : 0626
                 53   04   AC  D0 00073           MOVL     UCB, R3                                      : 0627
                           53  DD 00077           PUSHL    R3                                           :
                           7E  D4 00079           CLRL     -(SP)                                        :
        0000G    CF        02  FB 0007B           CALLS    #2, SEND_ERRLOG                              :
           00    BE      6248  D0 00080           MOVL     (UCBLIST)[NVOL], @CCB                        : 0629
                 18        59  E9 00085           BLBC     UNLOAD, 9$                                   : 0630
                           7E  7C 00088           CLRQ     -(SP)                                        : 0631
                           7E  7C 0008A           CLRQ     -(SP)                                        :
                           7E  7C 0008C           CLRQ     -(SP)                                        :
```

```
                        7E  7C 0008E          CLRQ    -(SP)                                   ;
                    7E  01  7D 00090          MOVQ    #1, -(SP)                               ;
                0000G   CF  DD 00093          PUSHL   IO_CHANNEL                              ;
                        7E  D4 00097          CLRL    -(SP)                                   ;
00000000G       00          FB 00099          CALLS   #12, SYS$QIOW                           ;
                        7E  7C 000A0 9$:      CLRQ    -(SP)                                   ; 0632
                        7E  7C 000A2          CLRQ    -(SP)                                   ;
                        7E  7C 000A4          CLRQ    -(SP)                                   ;
                        7E  7C 000A6          CLRQ    -(SP)                                   ;
                    7E  11  7D 000A8          MOVQ    #17, -(SP)                              ;
                0000G   CF  DD 000AB          PUSHL   IO_CHANNEL                              ;
                        7E  D4 000AF          CLRL    -(SP)                                   ;
00000000G       00          FB 000B1          CALLS   #12, SYS$QIOW                           ;
      08    AE        20  A3  D0 000B8          MOVL    32(R3), LKSTS+4                         ; 0637
                        2A  13 000BD          BEQL    11$                                     ;
                        2C  A3 D5 000BF          TSTL    44(R3)                                  ; 0638
                        25  12 000C2          BNEQ    11$                                     ;
                        7E  7C 000C4          CLRQ    -(SP)                                   ; 0645
                        7E  7C 000C6          CLRQ    -(SP)                                   ;
                        7E  7C 000C8          CLRQ    -(SP)                                   ;
                    7E  2A  7D 000CA          MOVQ    #42, -(SP)                              ;
                        24  AE 9F 000CD          PUSHAB  LKSTS                                   ;
                        04  DD 000D0          PUSHL   #4                                      ;
                        01  DD 000D2          PUSHL   #1                                      ;
00000000G       00          FB 000D4          CALLS   #11, SYS$ENQW                           ;
                    5A  50  D0 000DB          MOVL    R0, STS                                 ;
                    04  5A  E9 000DE          BLBC    STS, 10$                                ; 0646
                    04      04 AE E8 000E1          BLBS    LKSTS, 11$                              ; 0647
                          FEFF 000E5 10$:       BUGW                                            ; 0648
                         0000* 000E7          .WORD   <BUG$_XQPERR!4>                         ;
                    53      04 AC  D0 000E9 11$:      MOVL    UCB, R3                                 ; 0655
                    57      1C A3  D0 000ED          MOVL    28(R3), ORB                             ;
            0000G   CF          00 FB 000F1          CALLS   #0, LOCK_IODB                           ; 0656
              3A    A3    0228  8F  AA 000F6          BICW2   #552, 58(R3)                            ; 0660
                        34  A3  D4 000FC          CLRL    52(R3)                                  ; 0661
              66    A3        10  8A 000FF          BICB2   #16, 102(R3)                            ; 0662
                        5C  A3  B7 00103          DECW    92(R3)                                  ; 0663
                        18  A7  7C 00106          CLRQ    24(ORB)                                 ; 0664
                        20  A7  7C 00109          CLRQ    32(ORB)                                 ; 0666
                        67  D4 0010C          CLRL    (ORB)                                   ; 0668
                        08  AE  D5 0010E          TSTL    LKSTS+4                                 ; 0675
                        37  13 00111          BEQL    15$                                     ;
                        0C  AE 7C 00113          CLRQ    LKSTS+8                                 ; 0678
                        14  AE 7C 00116          CLRQ    LKSTS+16                                ; 0680
                        7E  7C 00119          CLRQ    -(SP)                                   ; 0690
                        7E  7C 0011B          CLRQ    -(SP)                                   ;
                        7E  7C 0011D          CLRQ    -(SP)                                   ;
                        7E  D4 0011F          CLRL    -(SP)                                   ;
                    7E  6B  8F 9A 00121          MOVZBL  #107, -(SP)                             ;
                        24  AE 9F 00125          PUSHAB  LKSTS                                   ;
                        2C  A3 D5 00128          TSTL    44(R3)                                  ;
                        04  13 0012B          BEQL    12$                                     ;
                        05  DD 0012D          PUSHL   #5                                      ;
                        02  11 0012F          BRB     13$                                     ;
                        01  DD 00131 12$:      PUSHL   #1                                      ;
                        01  DD 00133 13$:      PUSHL   #1                                      ;
00000000G       00          0B FB 00135          CALLS   #11, SYS$ENQ                            ;
```

```
                    5A              50   D0 0013C          MOVL    R0, STS                                 ; 0691
                    04              5A   E9 0013F          BLBC    STS, 14$                                ; 0692
                    04        04    AE   E8 00142          BLBS    LKSTS, 15$                              ; 0693
                                  FEFF   00146 14$:       BUGW                                            ;
                                  0000*  00148            .WORD   <BUG$_XQPERR!4>                         ;
                    55        04    AC   D0 0014A 15$:      MOVL    UCB, R5                                 ; 0699
                    54 00000000G    00   D0 0014E          MOVL    SCH$GL_CURPCB, R4                       ;
                       00000000G    00   16 00155          JSB     IOC$DALLOC_DMT                          ;
              0000G CF              00   FB 0015B          CALLS   #0, UNLOCK_IODB                         ; 0701
                    02              58   F4 00160 16$:      SOBGEQ  NVOL, 17$                               ; 0624
                                    03   11 00163          BRB     18$                                     ;
                                FF06   31 00165 17$:       BRW     8$                                      ;
                          20        A6   DD 00168 18$:      PUSHL   32(VCB)                                 ; 0706
              0000G CF              01   FB 0016B          CALLS   #1, DEALLOCATE                          ;
                          34        A6   DD 00170          PUSHL   52(VCB)                                 ; 0707
              0000G CF              01   FB 00173          CALLS   #1, DEALLOCATE                          ;
                    52        3C    B6   0F 00178 19$:      REMQUE  @60(VCB), PAGE                          ; 0714
                                    0B   1D 0017C          BVS     20$                                     ;
                                    01   DD 0017E          PUSHL   #1                                      ; 0716
                                    52   DD 00180          PUSHL   PAGE                                    ;
              0000G CF              02   FB 00182          CALLS   #2, RET_FREE_PAGE                       ;
                                    EF   11 00187          BRB     19$                                     ; 0711
                                    56   DD 00189 20$:      PUSHL   VCB                                     ; 0719
              0000G CF              01   FB 0018B          CALLS   #1, DEALLOCATE                          ;
                    50      0000G   CF   D0 00190          MOVL    QUEUE_HEAD, R0                          ; 0720
                              0B    A0   97 00195          DECB    11(R0)                                  ;
                                         04 00198          RET                                             ; 0596
              0000G CF              00   FB 00199 21$:      CALLS   #0, UNLOCK_IODB                         ; 0723
                                         04 0019E          RET                                             ; 0725

; Routine Size:  415 bytes,    Routine Base:  $LOCKEDC1$ + 0000

;  344          0726  1
;  345          0727  1 END
;  346          0728  1
;  347          0729  0 ELUDOM



;                                PSECT SUMMARY
;
;      Name                      Bytes                         Attributes
;
;  $LOCKEDC1$                        415  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                              Library Statistics
;                                        --------  Symbols  --------      Pages       Processing
;      File                              Total     Loaded   Percent       Mapped      Time
;
```

```
;  _$255$DUA28:[SYSLIB]LIB.L32;1          18619        45         0       1000          00:01.9



;                                       COMMAND QUALIFIERS

;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:CHKDMO/OBJ=OBJ$:CHKDMO MSRC$:CHKDMO/UPDATE=(ENH$:CHKDMO)

; Size:         415 code + 0 data bytes
; Run Time:         00:13.5
; Elapsed Time:     00:34.8
; Lines/CPU Min:     3230
; Lexemes/CPU-Min: 24177
; Memory Used:  181 pages
; Compilation Complete
```

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

CALDAYNO
LIS

SYSFAC
DAT

ACPCTR
LIS

ALLOCB
LIS

SYSMSG
LIS

CHKACC
LIS

MTAACP

ACCESS
LIS

MTAAACP
MAP

ACCFL
LIS

CALCTV
LIS

BLOCK
LIS

CLOSFL
LIS

MTADEF
B32 -

CHKDMO
LIS