I 3

```
   CCCCCCCCCCCC  LLL                 IIIIIIIII     UUU         UUU  TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCCC  LLL                 IIIIIIIII     UUU         UUU  TTTTTTTTTTTTTTT  LLL
   CCCCCCCCCCCC  LLL                 IIIIIIIII     UUU         UUU  TTTTTTTTTTTTTTT  LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
CCC              LLL                    III        UUU         UUU        TTT        LLL
   CCCCCCCCCCCC  LLLLLLLLLLLLLLL     IIIIIIIII     UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCCC  LLLLLLLLLLLLLLL     IIIIIIIII     UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
   CCCCCCCCCCCC  LLLLLLLLLLLLLLL     IIIIIIIII     UUUUUUUUUUUUUUU        TTT        LLLLLLLLLLLLLLL
```

```
                                                       H 16
**FILE**ID**SETVOLUME




  SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT  VV      VV    OOOOOO    LL          UU      UU  MM      MM  EEEEEEEEEE
  SSSSSSSS  EEEEEEEEEE  TTTTTTTTTT  VV      VV    OOOOOO    LL          UU      UU  MM      MM  EEEEEEEEEE
SS          EE              TT      VV      VV  OO      OO  LL          UU      UU  MMMM  MMMM  EE
SS          EE              TT      VV      VV  OO      OO  LL          UU      UU  MMMM  MMMM  EE
SS          EE              TT      VV      VV  OO      OO  LL          UU      UU  MM  MM  MM  EE
SS          EE              TT      VV      VV  OO      OO  LL          UU      UU  MM  MM  MM  EE
  SSSSSS    EEEEEEEE        TT      VV      VV  OO      OO  LL          UU      UU  MM      MM  EEEEEEEE
  SSSSSS    EEEEEEEE        TT      VV      VV  OO      OO  LL          UU      UU  MM      MM  EEEEEEEE
        SS  EE              TT      VV      VV  OO      OO  LL          UU      UU  MM      MM  EE
        SS  EE              TT      VV      VV  OO      OO  LL          UU      UU  MM      MM  EE
        SS  EE              TT        VV  VV    OO      OO  LL          UU      UU  MM      MM  EE              ....
        SS  EE              TT        VV  VV    OO      OO  LL          UU      UU  MM      MM  EE              ....
SSSSSSSS    EEEEEEEEEE      TT          VV        OOOOOO    LLLLLLLLLL  UUUUUUUUUU  MM      MM  EEEEEEEEEE      ....
SSSSSSSS    EEEEEEEEEE      TT          VV        OOOOOO    LLLLLLLLLL  UUUUUUUUUU  MM      MM  EEEEEEEEEE      ....

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
```

```
0
0 MODULE setvol   (
0                 IDENT = 'V04-000',
0                 ADDRESSING_MODE(EXTERNAL=GENERAL,
0                                 NONEXTERNAL=LONG_RELATIVE)
0                 ) =
1 BEGIN
1
1 !
1 !****************************************************************************
1 !*                                                                          *
1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                 *
1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                  *
1 !*  ALL RIGHTS RESERVED.                                                    *
1 !*                                                                          *
1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE   *
1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
1 !*  TRANSFERRED.                                                            *
1 !*                                                                          *
1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
1 !*  CORPORATION.                                                            *
1 !*                                                                          *
1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
1 !*                                                                          *
1 !*                                                                          *
1 !****************************************************************************
1 !
1
1 !++
1 ! FACILITY:      Set Volume Command
1 !
1 ! ABSTRACT:
1 !
1 !       This module processes the Set Volume command.
1 !
1 ! ENVIRONMENT:
1 !
1 !       Vax native, privileged user mode
1 !
1 !--
1 !
1 ! AUTHOR:        Gerry Smith              CREATION DATE: 3-Nov-1981
1 !
1 ! MODIFIED BY:
1 !
1 !
1 !       V03-010 AEW0003         Anne E. Warner          18-Jul-1984
1 !               Add a check to see if the device specified is a
1 !               Files-11 format disk and if not tell the user.
1 !               This check includes the new error message:
1 !                 set$_notdisk, device is not a files-11 format disk
1 !
```

```
1 !                  Also check to see if qualifiers with 'values' check
1 !                  that the qualifier is present before looking for values.
1 !                  This is because most qualifiers are negatable now.
1 !                  As a result this check was added to /LABEL when it is
1 !                  checked for.
1 !
1 !          V03-009 DAS0001          David Solomon          09-Jul-1984
1 !                  Add support for /REBUILD - perform volume rebuild.
1 !
1 !          V03-008 AEW0002          Anne E. Warner         24-May-1984
1 !                  Change RMS access to $QIOW access so that the home
1 !                  block can be found in ODS1 structure blocks.  The
1 !                  problem was that RMS sees the End-of-File as zero
1 !                  on an ODS1 initialized volume and will not look for a
1 !                  valid home block.
1 !
1 !          V03-007 LMP0221          L. Mark Pilant,        9-Apr-1984  10:46
1 !                  Change UCB$L_OWNUIC to ORB$L_OWNER and UCB$W_VPROT to
1 !                  ORB$W_PROT.
1 !
1 !          V03-006 MCN0164          Maria del C. Nasr      03-Apr-1984
1 !                  The /DATA_CHECK qualifier must accept NOREAD and NOWRITE.
1 !
1 !          V03-005 AEW0001          Anne E. Warner         21-Mar-1984
1 !                  Add a check to see if volume is mounted foreign.  If
1 !                  it is it cannot be modified because it is not in
1 !                  Files-11 format so notify the user and exit.
1 !
1 !          V03-004 GAS0132          Gerry Smith            13-May-1983
1 !                  Add [NO]HIGHWATER, [NO]UNLOAD, [NO]MOUNT_VERIFICATION,
1 !                  [NO]ERASE_ON_DELETE.  Also modify VOLSET.SYS on the
1 !                  root volume for volume sets if /LABEL specified.
1 !
1 !          V03-003 GAS0121          Gerry Smith            14-Apr-1983
1 !                  For ODS1 disks, fold long UICs into <377,377>.
1 !
1 !          V03-002 GAS0112          Gerry Smith            29-Mar-1983
1 !                  Convert to new CLI interface, and new command dispatcher.
1 !
1 !          V03-001 GAS52349         Gerry Smith            4-Jan-1983
1 !                  Remove one level of indirection from the DEVCHAR field
1 !                  of the UCB when modifying its contents.
1 !
1 !          V03-006 GAS0091          Gerry Smith            19-Oct-1982
1 !                  Change input request for new CLD syntax.
1 !
1 !          V03-005 GAS0040          Gerry Smith            2-Feb-1982
1 !                  Fix privilege checking to check for write access to
1 !                  the volume's index file.  Also, fix write bug that
1 !                  prevented modified home blocks to be written back.
1 !
1 !          V03-004 GAS0033          Gerry Smith            12-Jan-1982
1 !                  Fix various bugs.
1 !
1 !          V03-003 GAS0030          Gerry Smith            1-Jan-1982
1 !                  Add /RETENTION, the default retention period for files
1 !                  created on a volume.
```

```
1 !
1 !      V03-002 GAS0026         Gerry Smith             18-Dec-1981
1 !              Use shared message file, and lower fatal messages to
1 !              simple error messages.
1 !
1 !      V03-001 GAS0025         Gerry Smith             14-Dec-1981
1 !              Add /LOG qualifier
1 !
1 !**
```

```
LIBRARY 'SYS$LIBRARY:LIB';
LIBRARY 'SYS$LIBRARY:TPAMAC';


!
! ***** Note: The following macro violates the Bliss language definition
! ***** in that it makes use of the value of SP while building the arg list.
! ***** It is the opinion of the Bliss maintainers that this usage is safe
! ***** from planned future optimizations.
!
! Macro to call the change mode to kernel system service.
! Macro call format is "KERNEL_CALL (ROUTINE, ARG1, ARG2, ... )".
!
MACRO
        KERNEL_CALL (R) =
            BEGIN
            EXTERNAL ROUTINE SYS$CMKRNL : ADDRESSING_MODE (ABSOLUTE);
            BUILTIN SP;
            SYS$CMKRNL (R, .SP, %LENGTH-1
                        %IF %LENGTH GTR 1 %THEN ,%REMAINING %FI)
            END%;

```

```
1 FORWARD ROUTINE
1       set$volume : NOVALUE,                   ! Main routine for volume
1       get_quals,                              ! Get qualifiers
1       parse_class,                            ! Parse a protection class
1       process_volume_set : NOVALUE,           ! Process volume set
1       process_one_volume : NOVALUE,           ! Process each volume
1       modify_volset : NOVALUE,                ! Fix VOLSET.SYS
1       set_home,                               ! Modify the homeblock
1       set_ucbvcb : NOVALUE,                   ! Modify the UCB and VCB for the disk
1       read_homeblock;                         ! Find and read first good homeblock
1
1
1 EXTERNAL ROUTINE
1       cli$present,                            ! Get qualifier
1       cli$get_value,                          ! Get value for qualifier
1       lib$file_scan,                          ! Routine to get next directory
1       check_privilege : NOVALUE,              ! Routine to check for privilege
1       search_error,                           ! Where to go if file search fails
1       file_error,                             ! Where to go if file error occurs
1       checksum2,                              ! Compute checksum
1       get_channelucb,                         ! Routine to get address of UCB
1       lib$cvt_dtb,                            ! Convert decimal to number
1       lib$cvt_dtime,                          ! Convert delta time
1       lib$tparse,                             ! Parser
1       parse_uic,                              ! Parse a UIC
1       sys$fao;                                ! Formatted ASCII output
1
1 !
1 ! External data references
1 !
1 EXTERNAL
1 !
1 ! Data
1 !
1       exte_value,                                     ! EXTENSION value
1       uic_value,                                      ! Owner UIC
1       group,                                          ! UIC group number
1       member;                                         ! UIC member number
1
1 !
1 ! Error messages
1 !
1 EXTERNAL LITERAL
1       cli$_ivprot,                            ! Invalid protection value
1       cli$_absent,
1       set$_operreq,                           ! OPER privilege required
1       set$_badfrmt,                           ! Volume doesn't have Files-11 format
1       set$_hbread,                            ! Error reading homeblock
1       set$_hbwrite,                           ! Error writing homeblock
1       set$_modified,                          ! Volume modified
1       set$_nohome,                            ! Volume has no good home block
1       set$_notdisk,                           ! Device is not a files-11 format disk
1       set$_notmod,                            ! Volume not modified
1       set$_notods2,                           ! Qualifier invalid for ODS1
1       set$_readerr,                           ! Error reading volume
1       set$_sysnotupd,                         ! Error updating ucb and vcb
1       set$_writeerr;                          ! Could not write to file
```

```
;  205        0203   1
;  206        0204   1 !
;  207        0205   1 ! Declare some shared messages
;  208        0206   1 !
;  209      P 0207   1 $SHR_MSGDEF      (SET,119,LOCAL,
;  210      P 0208   1                  (valerr,          error),
;  211      P 0209   1                  (syntax,          error),
;  212      P 0210   1                  (openout,         error),
;  213      P 0211   1                  (closeout,        error),
;  214        0212   1                  (invquaval,       error));
;  215        0213   1
;  216        0214   1
;  217        0215   1 !
;  218        0216   1 ! Literal data definitions
;  219        0217   1 !
;  220        0218   1 LITERAL
;  221        0219   1     true = 1,
;  222        0220   1     false = 0;
;  223        0221   1
;  224        0222   1 LITERAL
;  225      P 0223   1     $EQULST
;  226      P 0224   1         (QUAL_,,1,1,
;  227      P 0225   1         (access,),                      ! ACCESSED bit
;  228      P 0226   1         (data,),                        ! DATA_CHECK bit
;  229      P 0227   1         (exte,),                        ! EXTENSION bit
;  230      P 0228   1         (fprot,),                       ! FILE_PROTECTION bit
;  231      P 0229   1         (label,),                       ! LABEL bit
;  232      P 0230   1         (log,),                         ! LOG bit
;  233      P 0231   1         (owner,),                       ! OWNER_UIC bit
;  234      P 0232   1         (retent,),                      ! RETENTION bit
;  235      P 0233   1         (username,),                    ! USER_NAME bit
;  236      P 0234   1         (vprot,),                       ! PROTECTION bit
;  237      P 0235   1         (windows,),                     ! WINDOWS bit
;  238      P 0236   1         (erase,),                       ! [NO]ERASE
;  239      P 0237   1         (erase_val),
;  240      P 0238   1         (fhw,),                         ! [NO]HIGHWATER
;  241      P 0239   1         (fhw_val,),
;  242      P 0240   1         (mntver,),                      ! [NO]MOUNT_VERIFICATION
;  243      P 0241   1         (mntver_val,),
;  244      P 0242   1         (unl,),                         ! [NO]UNLOAD
;  245      P 0243   1         (unl_val,),
;  246      P 0244   1         (rebuild,),                     ! [NO]REBUILD
;  247      P 0245   1         (rebuild_val,),
;  248        0246   1         (lbl_cpy,));                    ! Old label was saved
;  249        0247   1 LITERAL
;  250      P 0248   1     $EQULST
;  251      P 0249   1         (DATA_,,1,1,
;  252      P 0250   1         (read,),                                ! DATA_CHECK = READ
;  253      P 0251   1         (write,),                               ! DATA_CHECK = WRITE
;  254      P 0252   1         (noread,),                              ! DATA_CHECK = NOREAD
;  255        0253   1         (nowrite,));                            ! DATA_CHECK = NOWRITE
;  256        0254   1
```

```
1 !
1 ! Define storage for this module that must be global
1 !
1 GLOBAL
1     acc_value,                                  ! ACCESSED value
1     fprot_value,                                ! FILE_PROTECTION value
1     label_value : VECTOR[2],                    ! LABEL label
1     vprot_value,                                ! PROTECTION value
1     retmin_value : VECTOR[2],                   ! Minimum retention period
1     retmax_value : VECTOR[2],                   ! Maximum retention period
1     user_value : VECTOR[2],                     ! USER_NAME
1     window_value;                               ! WINDOWS
1
```

```
!
! Define own storage for this module
!
OWN
    flags : BITVECTOR[32],                      ! Qualifier flags word
    dflags : BITVECTOR[32],                     ! DATA_CHECK flags word
    user_label : VECTOR[12,BYTE],               ! Place to put username
    label_buff : VECTOR[vcb$s_volname,BYTE],      ! Place to store old label
    buffer : BLOCK[512,BYTE],                   ! Place for home block
    acc_inc : BYTE,                             ! Increment to LRU limit
    ods1 : BYTE,                                ! ODS1 indicator
    channel,                                    ! Channel for $QIOW

    result_file : VECTOR[nam$c_maxrss,BYTE],

    NAM : $NAM (RSA = result_file,
                RSS = nam$c_maxrss),

    FAB : $FAB (DNA = UPLIT BYTE ('[0,0]INDEXF.SYS'),
                DNS = %CHARCOUNT ('[0,0]INDEXF.SYS'),
                FAC = (get, put, bio),
                SHR = (get, upi),
                NAM = nam,
                FOP = ufo);
```

```
1 GLOBAL ROUTINE set$volume : NOVALUE =
1 !++
1 !
1 ! Functional description
1 !
1 !       This is the main control module for SET VOLUME.  It obtains the
1 !       qualifiers and, for each volume specification, calls the routine
1 !       that actually modifies the volume's home block.
1 !
1 ! Calling sequence
1 !
1 !       CALL set$volume()
1 !
1 ! Input parameters
1 !       none
1 !
1 ! Output parameters
1 !       none
1 !
1 ! Implicit outputs
1 !       none
1 !
1 ! Routine value
1 !       none
1 !
1 ! Side effects
1 !       none
1 !
1 !--
2 BEGIN
2
2 LOCAL
2     dyn_desc : $BBLOCK[dsc$c_s_bln];
2
2 !
2 ! Check that the image is running with appropriate privilege.
2 !
2 check_privilege();
2
2 !
2 ! Get the command qualifiers.
2 !
2 IF NOT get_quals()
2 THEN RETURN;
2
2
2 !
2 ! For each volume specified, perform the operations requested.
2 !
2 $init_dyndesc(dyn_desc);                                 ! Make desc. dynamic
2 WHILE cli$get_value(%ASCID 'VOLUME', dyn_desc)          ! For each volume specified,
2 DO
3     BEGIN
3     LOCAL
3         status,
3         max_rvn : volatile,                               ! Total volumes in set
3         original_rvn : volatile,                          ! Original rvn (this disk)
```

```
  3         root_desc : VECTOR[2],                            ! Root volume descriptor
  3         root_buffer : VECTOR[128,BYTE],                   ! Place to put root name
  3         iosb : VECTOR[4,WORD],                            ! Status block for GETDVI
  3         devchar : $BBLOCK [DIB$K_LENGTH],                 ! Longword of device characterisitcs
  3                                                           ! defined in $DEVDEF
  3         dvi_list : $ITMLST_DECL(ITEMS=4);                 ! $GETDVI item list
  3
  3 !
  3 ! Get the root volume, the total number of volumes, and the volume number of
  3 ! the original volume.
  3 !
P 3     $ITMLST_INIT(ITMLST = dvi_list,                       ! Set up DVI list
P 3                 (ITMCOD = dvi$_rootdevnam,                ! Want root volume
P 3                  BUFADR = root_buffer,                    ! name,
P 3                  BUFSIZ = %ALLOCATION(root_buffer),
P 3                  RETLEN = root_desc),
P 3                 (ITMCOD = dvi$_volnumber,                 ! this disk's volume
P 3                  BUFADR = original_rvn),                  ! number, and
P 3                 (ITMCOD = dvi$_volcount,                  ! the total number of
P 3                  BUFADR = max_rvn),                       ! volumes.
P 3                 (ITMCOD = dvi$_devchar,                   ! Get the device characteristics
  3                  BUFADR = devchar));                      ! to find if mounted foreign.
  3
  3     root_desc[1] = root_buffer;                           ! Set up parameter for
  3                                                           ! later processing.
P 3     status = $GETDVIW(ITMLST = dvi_list,                  ! Get the information.
P 3                       DEVNAM = dyn_desc,
  3                       IOSB   = iosb);
  3     IF .status
  3     THEN status = .iosb[0];
  3     IF NOT .status                                        ! If a problem, signal,
  3     THEN SIGNAL(.status)                                  ! otherwise
  3     ELSE
  3 !
  3 !  If the device specified is not a Files-11 volume or the volume was mounted
  3 !  foreign it cannot be modified, so signal an error and exit.
  3 !
  3 !
  3 !  Check if a Files-11 volume was specified
  3 !
  4       BEGIN
  4         IF NOT .devchar[dev$v_rnd]
  4         THEN
  5         BEGIN
  5         LOCAL
  5           nodisk_desc : $BBLOCK[dsc$c_s_bln];             ! descriptor for device
  5
  5           $INIT_DYNDESC (nodisk_desc);
  5           nodisk_desc[dsc$w_length] = .root_desc[0];      ! length of device name
  5           nodisk_desc[dsc$a_pointer] = .root_desc[1];     ! device name
  5           SIGNAL                                          ! inform user of error
  5            (set$_notmod, 1, nodisk_desc, set$_notdisk);
  5           RETURN false;
  4         END;
  4 !
  4 !  It is a Files-11 device so check if mounted foreign
  4 !
```

```
;  412          0407  4          IF .devchar[dev$v_for]
;  413          0408  4          THEN
;  414          0409  5          BEGIN
;  415          0410  5          LOCAL
;  416          0411  5             foreign_desc : $BBLOCK[dsc$c_s_bln];             ! descriptor for volume name
;  417          0412  5
;  418          0413  5             $INIT_DYNDESC (foreign_desc);
;  419          0414  5             foreign_desc[dsc$w_length] = .root_desc[0];     ! length of volume name
;  420          0415  5             foreign_desc[dsc$a_pointer] = .root_desc[1];    ! volume name
;  421          0416  5             SIGNAL(                                         ! inform user of error
;  422          0417  5              (set$_notmod, 1, foreign_desc, set$_badfrmt);
;  423          0418  5             RETURN false;
;  424          0419  4          END;
;  425          0420  4  !
;  426          0421  4  !  If everything is alright process the volume set.
;  427          0422  4  !
;  428          0423  4          process_volume_set(root_desc,
;  429          0424  4                             .original_rvn,
;  430          0425  4                             .max_rvn);
;  431          0426  3        END;
;  432          0427  2      END;
;  433          0428  2
;  434          0429  2  RETURN;
;  435          0430  1  END;


                                                                        .TITLE  SETVOL
                                                                        .IDENT  \V04-000\

                                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

53  59  53  2E  46  58  45  44  4E  49  5D  30  2C  30  5B  00000 P.AAA:  .ASCII  \[0,0]INDEXF.SYS\                            ;
                                                            0000F         .BLKB   1
                            00  00  45  4D  55  4C  4F  56  00010 P.AAC:  .ASCII  \VOLUME\<0><0>                               ;
                                                  010E0006  00018 P.AAB:  .LONG   17694726                                     ;
                                                 00000000'  0001C         .ADDRESS P.AAC                                       ;

                                                                        .PSECT  $OWN$,NOEXE,2

                                                            00000 FLAGS:  .BLKB   4
                                                            00004 DFLAGS: .BLKB   4
                                                            00008 USER_LABEL:
                                                                          .BLKB   12
                                                            00014 LABEL_BUFF:
                                                                          .BLKB   12
                                                            00020 BUFFER: .BLKB   512
                                                            00220 ACC_INC:.BLKB   1
                                                            00221 ODST:   .BLKB   1
                                                            00222         .BLKB   2
                                                            00224 CHANNEL:.BLKB   4
                                                            00228 RESULT_FILE:
                                                                          .BLKB   255
                                                            00327         .BLKB   1
                                                        02  00328 NAM:    .BYTE   2                                            ;
                                                        60  00329         .BYTE   96                                           ;
                                                        FF  0032A         .BYTE   -1                                           ;
                                                        00  0032B         .BYTE   0                                            ;
```

```
00000000'  0032C          .ADDRESS RESULT_FILE
      00   00330          .BYTE    0
      00   00331          .BYTE    0
      00   00332          .BYTE    0
      00   00333          .BYTE    0
00000000   00334          .LONG    0
00000000   00338          .LONG    0
   0000#   0033C          .WORD    0[8]
   0000#   0034C          .WORD    0[3]
   0000#   00352          .WORD    0[3]
00000000   00358          .LONG    0
00000000   0035C          .LONG    0
      00   00360          .BYTE    0
      00   00361          .BYTE    0
      00   00362          .BYTE    0
      00   00363          .BYTE    0
      00   00364          .BYTE    0
      00   00365          .BYTE    0
     00#   00366          .BYTE    0[2]
00000000   00368          .LONG    0
00000000   0036C          .LONG    0
00000000   00370          .LONG    0
00000000   00374          .LONG    0
00000000   00378          .LONG    0
00000000   0037C          .LONG    0
00000000#  00380          .LONG    0[2]
      03   00388 FAB:     .BYTE    3
      50   00389          .BYTE    80
    0000   0038A          .WORD    0
00020000   0038C          .LONG    131072
00000000   00390          .LONG    0
00000000   00394          .LONG    0
00000000   00398          .LONG    0
    0000   0039C          .WORD    0
      23   0039E          .BYTE    35
      42   0039F          .BYTE    66
00000000   003A0          .LONG    0
      00   003A4          .BYTE    0
      00   003A5          .BYTE    0
      00   003A6          .BYTE    0
      02   003A7          .BYTE    2
00000000   003A8          .LONG    0
00000000   003AC          .LONG    0
00000000'  003B0          .ADDRESS NAM
00000000   003B4          .LONG    0
00000000'  003B8          .ADDRESS P.AAA
      00   003BC          .BYTE    0
      0F   003BD          .BYTE    15
    0000   003BE          .WORD    0
00000000   003C0          .LONG    0
    0000   003C4          .WORD    0
      00   003C6          .BYTE    0
      00   003C7          .BYTE    0
00000000   003C8          .LONG    0
00000000   003CC          .LONG    0
    0000   003D0          .WORD    0
      00   003D2          .BYTE    0
```

```
                          00  003D3           .BYTE    0                                          ;
                    00000000  003D4           .LONG    0                                          ;

                                              .PSECT   $GLOBAL$,NOEXE,2

                              00000 ACC_VALUE::
                                              .BLKB    4
                              00004 FPROT_VALUE::
                                              .BLKB    4
                              00008 LABEL_VALUE::
                                              .BLKB    8
                              00010 VPROT_VALUE::
                                              .BLKB    4
                              00014 RETMIN_VALUE::
                                              .BLKB    8
                              0001C RETMAX_VALUE::
                                              .BLKB    8
                              00024 USER_VALUE::
                                              .BLKB    8
                              0002C WINDOW_VALUE::
                                              .BLKB    4

                                              .EXTRN   CLI$PRESENT, CLI$GET_VALUE
                                              .EXTRN   LIB$FILE_SCAN, CHECK_PRIVILEGE
                                              .EXTRN   SEARCH_ERROR, FILE_ERROR
                                              .EXTRN   CHECKSUM2, GET_CHANNELUCB
                                              .EXTRN   LIB$CVT_DTB, LIB$CVT_DTIME
                                              .EXTRN   LIB$TPARSE, PARSE_UIC
                                              .EXTRN   SYS$FAO, EXTE_VALUE
                                              .EXTRN   UIC_VALUE, GROUP
                                              .EXTRN   MEMBER, CLI$_IVPROT
                                              .EXTRN   CLI$_ABSENT, SET$_OPERREQ
                                              .EXTRN   SET$_BADFRMT, SET$_HBREAD
                                              .EXTRN   SET$_HBWRITE, SET$_MODIFIED
                                              .EXTRN   SET$_NOHOME, SET$_NOTDISK
                                              .EXTRN   SET$_NOTMOD, SET$_NOTODS2
                                              .EXTRN   SET$_READERR, SET$_SYSNOTUPD
                                              .EXTRN   SET$_WRITEERR, SYS$GETDVIW

                                              .PSECT   $CODE$,NOWRT,2

                           0004 00000         .ENTRY   SET$VOLUME, Save R2                        ; 0293
      52 00000000G      00   9E 00002         MOVAB    LIB$SIGNAL, R2                             ;
      5E      FEB0      CE   9E 00009         MOVAB    -336(SP), SP                               ;
00000000G 00            00   FB 0000E         CALLS    #0, CHECK_PRIVILEGE                        ; 0330
00000000V EF            00   FB 00015         CALLS    #0, GET_QUALS                              ; 0335
      1B                50   E9 0001C         BLBC     R0, 2$                                     ;
   F8 AD 020E0000       8F   D0 0001F         MOVL     #34471936, DYN_DESC                        ; 0342
              FC        AD   D4 00027         CLRL     DYN_DESC+4                                 ;
              F8        AD   9F 0002A 1$:     PUSHAB   DYN_DESC                                   ; 0343
        00000000'       EF   9F 0002D         PUSHAB   P.AAB                                      ;
00000000G 00            02   FB 00033         CALLS    #2, CLI$GET_VALUE                          ;
      01                50   E8 0003A 2$:     BLBS     R0, 3$                                     ;
                        04      0003D         RET                                                 ;
      50        08      AE   9E 0003E 3$:     MOVAB    DVI_LIST, $$ITMBLKPTR                      ; 0371
      80 00320080       8F   D0 00042         MOVL     #3276928, ($$ITMBLKPTR)+                   ;
      80      FF68      CD   9E 00049         MOVAB    ROOT_BUFFER, ($$ITMBLKPTR)+                ;
```

```
                     80       E8   AD  9E 0004E          MOVAB    ROOT_DESC, ($$ITMBLKPTR)+                  ;
                     80 002E0004   8F  D0 00052          MOVL     #3014660, ($$ITMBLKPTR)+                   ;
                     80       F0   AD  9E 00059          MOVAB    ORIGINAL_RVN, ($$ITMBLKPTR)+               ;
                                   80  D4 0005D          CLRL     ($$ITMBLKPTR)+                             ;
                     80 00300004   8F  D0 0005F          MOVL     #3145732, ($$ITMBLKPTR)+                   ;
                     80       F4   AD  9E 00066          MOVAB    MAX_RVN, ($$ITMBLKPTR)+                    ;
                                   80  D4 0006A          CLRL     ($$ITMBLKPTR)+                             ;
                     80 00020004   8F  D0 0006C          MOVL     #131076, ($$ITMBLKPTR)+                    ;
                     80       3C   AE  9E 00073          MOVAB    DEVCHAR, ($$ITMBLKPTR)+                    ;
                                   80  7C 00077          CLRQ     ($$ITMBLKPTR)+                             ;
                EC   AD     FF68   CD  9E 00079          MOVAB    ROOT_BUFFER, ROOT_DESC+4                   ; 0373
                                   7E  7C 0007F          CLRQ     -(SP)                                      ; 0377
                                   7E  D4 00081          CLRL     -(SP)                                      ;
                            FF60   CD  9F 00083          PUSHAB   IOSB                                       ;
                              18   AE  9F 00087          PUSHAB   DVI_LIST                                   ;
                              F8   AD  9F 0008A          PUSHAB   DYN_DESC                                   ;
                                   7E  7C 0008D          CLRQ     -(SP)                                      ;
          00000000G  00            08  FB 0008F          CALLS    #8, SYS$GETDVIW                            ;
                     08            50  E9 00096          BLBC     STATUS, 4$                                 ; 0378
                     50     FF60   CD  3C 00099          MOVZWL   IOSB, STATUS                               ; 0379
                     07            50  E8 0009E          BLBS     STATUS, 6$                                 ; 0380
                                   50  DD 000A1  4$:     PUSHL    STATUS                                     ; 0381
                     62            01  FB 000A3          CALLS    #1, LIB$SIGNAL                             ;
                                   82  11 000A6  5$:     BRB      1$                                         ;
   1B           3F   AE            04  E0 000A8  6$:     BBS      #4, DEVCHAR+3, 7$                          ; 0391
                     6E 020E0000   8F  D0 000AD          MOVL     #34471936, NODISK_DESC                     ; 0397
                              04   AE  D4 000B4          CLRL     NODISK_DESC+4                              ;
                     6E       E8   AD  B0 000B7          MOVW     ROOT_DESC, NODISK_DESC                     ; 0398
                04   AE       EC   AD  D0 000BB          MOVL     ROOT_DESC+4, NODISK_DESC+4                 ; 0399
                       00000000G   8F  DD 000C0          PUSHL    #SET$_NOTDISK                              ; 0401
                                   1D  11 000C6          BRB      8$                                         ;
                     28       3F   AE  E9 000C8  7$:     BLBC     DEVCHAR+3, 9$                              ; 0407
                     6E 020E0000   8F  D0 000CC          MOVL     #34471936, FOREIGN_DESC                    ; 0413
                              04   AE  D4 000D3          CLRL     FOREIGN_DESC+4                             ;
                     6E       E8   AD  B0 000D6          MOVW     ROOT_DESC, FOREIGN_DESC                    ; 0414
                04   AE       EC   AD  D0 000DA          MOVL     ROOT_DESC+4, FOREIGN_DESC+4                ; 0415
                       00000000G   8F  DD 000DF          PUSHL    #SET$_BADFRMT                              ; 0417
                              04   AE  9F 000E5  8$:     PUSHAB   FOREIGN_DESC                               ;
                                   01  DD 000E8          PUSHL    #1                                         ;
                       00000000G   8F  DD 000EA          PUSHL    #SET$_NOTMOD                               ;
                     62            04  FB 000F0          CALLS    #4, LIB$SIGNAL                             ;
                                       04 000F3          RET                                                 ; 0418
                              F4   AD  DD 000F4  9$:     PUSHL    MAX_RVN                                    ; 0425
                              F0   AD  DD 000F7          PUSHL    ORIGINAL_RVN                               ; 0424
                              E8   AD  9F 000FA          PUSHAB   ROOT_DESC                                  ; 0423
          00000000V  EF            03  FB 000FD          CALLS    #3, PROCESS_VOLUME_SET                     ;
                                   A0  11 00104          BRB      5$                                         ; 0343
                                       04 00106          RET                                                 ; 0430

; Routine Size:  263 bytes,    Routine Base:  $CODE$ + 0000
```

```
  1 ROUTINE get_quals =
  1 !++
  1 !
  1 ! This routine interrogates the CLI to get all the qualifiers and
  1 ! values.
  1 !
  1 !--
  2 BEGIN
  2
  2 BUILTIN
  2     addm,
  2     cmpm;
  2
  2 LOCAL
  2     status,
  2     desc : $BBLOCK[dsc$c_s_bln];
  2
  2 $init_dyndesc(desc);                                  ! Make the desc. dynamic
  2
  2 !
  2 ! /ACCESSED
  2 !
  2 IF cli$present(%ASCID 'ACCESSED')
  2 THEN
  3     BEGIN
  3     LOCAL privs : $BBLOCK[8];           ! Place to store the process privileges
  3     flags[qual_access] = 1;
  3
  3 !
  3 ! Call $SETPRV to get the current privileges of the process.  If the process
  3 ! does not have OPER, then signal an error and stop.
  3 !
P 4     IF NOT (status = $SETPRV(ENBFLG = 1,                   ! Enable
P 4                              PRVADR = 0,                   ! No new privileges
P 4                              PRMFLG = 1,                   ! Get current privileges
  4                              PRVPRV = privs))
  3     THEN
  4         BEGIN
  4         SIGNAL(.status);
  4         RETURN false;
  3         END;
  3     IF NOT .privs[prv$v_oper]
  3     THEN
  4         BEGIN
  4         SIGNAL(set$_operreq);
  4         RETURN false;
  3         END;
  3
  3 !
  3 ! The process has the correct privilege, so go ahead and get the value
  3 !
  3     acc_value = 3;                      ! Set up the default
  3
  3 !
  3 ! If a value was specified, use it; otherwise, use the default.
  3 !
  3     IF cli$get_value(%ASCID 'ACCESSED', desc)
```

```
3        THEN
4            BEGIN
4            IF NOT LIB$CVT_DTB(.desc[dsc$w_length],
4                               .desc[dsc$a_pointer],
4                               acc_value)
4            THEN
5                BEGIN
5                SIGNAL(set$_syntax, 1, desc);
5                RETURN false;
4                END;
4            IF .acc_value LSS 0                         ! Check that value is in range
4            OR .acc_value GTR 255
4            THEN
5                BEGIN
5                SIGNAL(set$_syntax, 1, desc, set$_valerr);
5                RETURN false;
4                END;
3            END;
2        END;
2    !
2    ! /DATA_CHECK
2    !
2    IF cli$present(%ASCID 'DATA_CHECK')
2    THEN
3        BEGIN
3        flags[qual_data] = 1;
3        IF NOT cli$get_value(%ASCID 'DATA_CHECK', desc)
3        THEN
3            dflags[data_write] = 1
3        ELSE
3            WHILE cli$get_value(%ASCID 'DATA_CHECK', desc) DO
4            BEGIN
4            IF CH$EQL(.desc[dsc$w_length], .desc[dsc$a_pointer],
4                      .desc[dsc$w_length], UPLIT(BYTE('WRITE')))
4            THEN dflags[data_write] = 1
4            ELSE IF CH$EQL(.desc[dsc$w_length], .desc[dsc$a_pointer],
4                           .desc[dsc$w_length], UPLIT(BYTE('READ')))
4            THEN dflags[data_read] = 1
4            ELSE IF CH$EQL(.desc[dsc$w_length], .desc[dsc$a_pointer],
4                           .desc[dsc$w_length], UPLIT(BYTE('NOWRITE')))
4            THEN dflags[data_nowrite] = 1
4            ELSE IF CH$EQL(.desc[dsc$w_length], .desc[dsc$a_pointer],
4                           .desc[dsc$w_length], UPLIT(BYTE('NOREAD')))
4            THEN dflags[data_noread] = 1
4            ELSE
5                BEGIN
5                SIGNAL(set$_syntax, 1, desc);
5                RETURN false;
4                END;
3            END;
2        END;
2    !
2    ! /[NO]ERASE_ON_DELETE
2    !
2    status = cli$present(%ASCID 'ERASE_ON_DELETE');
```

```
2 IF .status NEQ cli$_absent
2 THEN
3     BEGIN
3     flags[qual_erase] = 1;
3     flags[qual_erase_val] = .status;
2     END;
2
2 !
2 ! /EXTENSION
2 !
2 IF cli$present(%ASCID 'EXTENSION')
2 THEN
3     BEGIN
3     flags[qual_exte] = 1;
3     exte_value = 5;
3     IF cli$get_value(%ASCID 'EXTENSION', desc)
3     THEN
4         BEGIN
4         IF NOT lib$cvt_dtb(.desc[dsc$w_length],
4                            .desc[dsc$a_pointer],
4                            exte_value)
4         THEN
5             BEGIN
5             SIGNAL(set$_syntax, 1, desc);
5             RETURN false;
4             END;
4         IF .exte_value LSS 0
4         OR .exte_value GTR 65535
4         THEN
5             BEGIN
5             SIGNAL(set$_syntax, 1, desc, set$_valerr);
5             RETURN false;
4             END;
3         END;
2     END;
2
2 !
2 ! /FILE_PROTECTION
2 !
2 IF cli$present(%ASCID 'FILE_PROTECTION')
2 THEN
3     BEGIN
3     BIND
3         setpro_mask = fprot_value + 2 : WORD,
3         setpro_prot = fprot_value : WORD;
3
3     flags[qual_fprot] = 1;
3     fprot_value = 0;
3
3     IF cli$present(%ASCID 'FILE_PROTECTION.SYSTEM')
3     THEN
4         BEGIN
4         setpro_mask = .setpro_mask OR %X'000F';
4         IF cli$get_value(%ASCID 'FILE_PROTECTION.SYSTEM',desc)
4         THEN setpro_prot = parse_class(desc);
3         END;
3     IF cli$present(%ASCID 'FILE_PROTECTION.OWNER')
```

```
3          THEN
4              BEGIN
4              setpro_mask = .setpro_mask OR %X'00F0';
4              IF cli$get_value(%ASCID 'FILE_PROTECTION.OWNER',desc)
4              THEN setpro_prot = .setpro_prot OR parse_class(desc)^4;
3              END;
3          IF cli$present(%ASCID 'FILE_PROTECTION.GROUP')
3          THEN
4              BEGIN
4              setpro_mask = .setpro_mask OR %X'0F00';
4              IF cli$get_value(%ASCID 'FILE_PROTECTION.GROUP',desc)
4              THEN setpro_prot = .setpro_prot OR parse_class(desc)^8;
3              END;
3          IF cli$present(%ASCID 'FILE_PROTECTION.WORLD')
3          THEN
4              BEGIN
4              setpro_mask = .setpro_mask OR %X'F000';
4              IF cli$get_value(%ASCID 'FILE_PROTECTION.WORLD',desc)
4              THEN setpro_prot = .setpro_prot OR parse_class(desc)^12;
3              END;
2          END;
2
2      !
2      ! /[NO]HIGHWATER_MARKING
2      !
2      status = cli$present(%ASCID 'HIGHWATER_MARKING');
2      IF .status NEQ cli$_absent
2      THEN
3          BEGIN
3          flags[qual_fhw] = 1;
3          flags[qual_fhw_val] = NOT .status;
2          END;
2
2      !
2      ! /LABEL
2      !
2      IF cli$present(%ASCID 'LABEL')
2      THEN
2          IF cli$get_value(%ASCID 'LABEL', desc)
2          THEN
3          BEGIN
3          flags[qual_label] = 1;
3          IF .desc[dsc$w_length] GTR vcb$s_volname
3          THEN
4              BEGIN
4              SIGNAL(set$_syntax, 1, desc);
4              RETURN false;
3              END;
3          label_value[0] = .desc[dsc$w_length];
3          label_value[1] = .desc[dsc$a_pointer];
3          $init_dyndesc(desc);
2          END;
2
2      !
2      ! /LOG
2      !
2      flags[qual_log] = cli$present(%ASCID 'LOG');
```

```
  2
  2 !
  2 ! /[NO]MOUNT_VERIFICATION
  2 !
  2 status = cli$present(%ASCID 'MOUNT_VERIFICATION');
  2 IF .status NEQ cli$_absent
  2 THEN
  3     BEGIN
  3     flags[qual_mntver] = 1;
  3     flags[qual_mntver_val] = .status;
  2     END;
  2
  2 !
  2 ! /OWNER_UIC
  2 !
  2 IF cli$present(%ASCID 'OWNER_UIC')
  2 THEN
  3     BEGIN
  3     flags[qual_owner] = 1;
  3     IF NOT cli$get_value(%ASCID 'OWNER_UIC', desc)
  3     THEN
  4         BEGIN
  4         LOCAL
  4             iosb : VECTOR[4,WORD];
P 4         status = $GETJPIW(ITMLST = UPLIT(WORD(4,jpi$_uic),
P 4                                          uic_value,
P 4                                          0,
P 4                                          0),
  4                             IOSB = iosb);
  4         IF .status
  4         THEN status = .iosb[0];
  4         IF NOT .status
  4         THEN
  5             BEGIN
  5             SIGNAL(.status);
  5             RETURN false;
  4             END;
  4         END
  3     ELSE parse_uic(desc, uic_value);
  2     END;
  2
  2 !
  2 ! /PROTECTION
  2 !
  2 IF cli$present(%ASCID 'PROTECTION')
  2 THEN
  3     BEGIN
  3     BIND
  3         setpro_mask = vprot_value + 2 : WORD,
  3         setpro_prot = vprot_value : WORD;
  3
  3     flags[qual_vprot] = 1;
  3     vprot_value = 0;
  3
  3     IF cli$present(%ASCID 'PROTECTION.SYSTEM')
  3     THEN
  4         BEGIN
```

```
4               setpro_mask = .setpro_mask OR %X'000F';
4               IF cli$get_value(%ASCID 'PROTECTION.SYSTEM',desc)
4               THEN setpro_prot = parse_class(desc);
3               END;
3           IF cli$present(%ASCID 'PROTECTION.OWNER')
3           THEN
4               BEGIN
4               setpro_mask = .setpro_mask OR %X'00F0';
4               IF cli$get_value(%ASCID 'PROTECTION.OWNER',desc)
4               THEN setpro_prot = .setpro_prot OR parse_class(desc)^4;
3               END;
3           IF cli$present(%ASCID 'PROTECTION.GROUP')
3           THEN
4               BEGIN
4               setpro_mask = .setpro_mask OR %X'0F00';
4               IF cli$get_value(%ASCID 'PROTECTION.GROUP',desc)
4               THEN setpro_prot = .setpro_prot OR parse_class(desc)^8;
3               END;
3           IF cli$present(%ASCID 'PROTECTION.WORLD')
3           THEN
4               BEGIN
4               setpro_mask = .setpro_mask OR %X'F000';
4               IF cli$get_value(%ASCID 'PROTECTION.WORLD',desc)
4               THEN setpro_prot = .setpro_prot OR parse_class(desc)^12;
3               END;
2           END;
2
2   !
2   ! /[NO]REBUILD
2   !
2   status = cli$present(%ASCID 'REBUILD');
2   IF .status NEQ cli$_absent
2   THEN
3       BEGIN
3       flags[qual_rebuild] = 1;
3       flags[qual_rebuild_val] = .status;
2       END;
2
2   !
2   ! /RETENTION
2   !
2   IF cli$present(%ASCID 'RETENTION')
2   THEN
3       BEGIN
3       LOCAL temp_desc : VECTOR[2];
3
3       flags[qual_retent] = 1;
3
3       CH$FILL(0, 8, retmin_value);                    ! Zero minimum value
3       CH$FILL(0, 8, retmax_value);                    ! Zero maximum value
3
3   !
3   ! If a minimum value was not supplied, signal an error
3   !
3       IF NOT cli$get_value(%ASCID 'RETENTION', desc)
3       THEN
4           BEGIN
```

```
4          SIGNAL(set$_syntax, 1, desc);
4          RETURN false;
3          END;
3
3 !
3 ! Convert the minimum retention value to 64-bit system delta time format
3 !
4      IF NOT (status = LIB$CVT_DTIME(desc, temp_desc))
3      THEN
4          BEGIN
4          SIGNAL(set$_syntax, 1, retmin_value);
4          RETURN false;
4          END
3      ELSE CH$MOVE(8, temp_desc, retmin_value);   ! If no error, put 64-bit
3                                                  ! delta time in place
3
3 !
3 ! If a maximum value was supplied, then convert it in the same way.
3 !
3      IF cli$get_value(%ASCID 'RETENTION', desc)
3      THEN
4          BEGIN
5          IF NOT (status = LIB$CVT_DTIME(desc, temp_desc))
4          THEN
5              BEGIN
5              SIGNAL(set$_syntax, 1, retmax_value);
5              RETURN false;
5              END
4          ELSE CH$MOVE(8, temp_desc, retmax_value);
4          END
4
4 !
4 ! If no maximum value was supplied, then use the lesser of:
4 !        twice the minimum value or
4 !        the minimum value plus one week
4 !
4 !
3      ELSE
4          BEGIN
4          LOCAL
4              double : VECTOR[2],                      ! Place for 2*RETMIN
4              week_plus : VECTOR[2],                   ! Place for RETMIN + 7
4              one_week : VECTOR[2]
4                  INITIAL(%X'D71BC000',                ! Binary for 7 days
4                          %X'FFFFFA7F');
4          ADDM(2, retmin_value, retmin_value, double);   ! Get 2*RETMIN
4          ADDM(2, one_week, retmin_value, week_plus);    ! and RETMIN+7
4          IF CMPM(2, double, week_plus) GTR 0            ! compare...
4          THEN CH$MOVE(8, double, retmax_value)
4          ELSE CH$MOVE(8, week_plus, retmax_value);
3          END;
2      END;
2
2 !
2 ! /[NO]UNLOAD
2 !
2 status = cli$present(%ASCID 'UNLOAD');
```

```
;  836          0830  2 IF .status NEQ cli$_absent
;  837          0831  2 THEN
;  838          0832  3     BEGIN
;  839          0833  3     flags[qual_unl] = 1;
;  840          0834  3     flags[qual_unl_val] = .status;
;  841          0835  2     END;
;  842          0836  2
;  843          0837  2 !
;  844          0838  2 ! /USER_NAME
;  845          0839  2 !
;  846          0840  2 IF cli$present(%ASCID 'USER_NAME')
;  847          0841  2 THEN
;  848          0842  3     BEGIN
;  849          0843  3     flags[qual_username] = 1;
;  850          0844  3     IF NOT cli$get_value(%ASCID 'USER_NAME', desc)
;  851          0845  3     THEN
;  852          0846  4         BEGIN
;  853          0847  4         LOCAL
;  854          0848  4             jpi_list : $ITMLST_DECL (ITEMS = 1),
;  855          0849  4             iosb : VECTOR[4,WORD];
;  856        P 0850  4         $ITMLST_INIT(ITMLST = jpi_list,
;  857        P 0851  4                     (ITMCOD = jpi$_username,
;  858        P 0852  4                      BUFADR = user_label,
;  859        P 0853  4                      BUFSIZ = hm2$s_ownername,
;  860          0854  4                      RETLEN = user_value[0]));
;  861        P 0855  4         status = $GETJPIW(ITMLST = jpi_list,
;  862          0856  4                           IOSB   = iosb);
;  863          0857  4         IF .status
;  864          0858  4         THEN status = .iosb[0];
;  865          0859  4         IF NOT .status
;  866          0860  4         THEN
;  867          0861  5             BEGIN
;  868          0862  5             SIGNAL(.status);
;  869          0863  5             RETURN false;
;  870          0864  4             END;
;  871          0865  4         user_value[1] = user_label;
;  872          0866  4         END
;  873          0867  3     ELSE
;  874          0868  4         BEGIN
;  875          0869  4         IF .desc[dsc$w_length] GTR hm2$s_ownername
;  876          0870  4         THEN
;  877          0871  5             BEGIN
;  878          0872  5             SIGNAL(set$_syntax, 1, desc);
;  879          0873  5             RETURN false;
;  880          0874  4             END;
;  881          0875  4         user_value[0] = .desc[dsc$w_length];
;  882          0876  4         user_value[1] = .desc[dsc$a_pointer];
;  883          0877  4         $init_dyndesc(desc);
;  884          0878  3         END;
;  885          0879  2     END;
;  886          0880  2
;  887          0881  2 !
;  888          0882  2 ! /WINDOWS
;  889          0883  2 !
;  890          0884  2 IF cli$present(%ASCID 'WINDOWS')
;  891          0885  2 THEN
;  892          0886  3     BEGIN
;  893          0887  3         flags[qual_windows] = 1;
;  894          0888  3         window_value = 7;
;  895          0889  3         IF cli$get_value(%ASCID 'WINDOWS', desc)
;  896          0890  3         THEN
;  897          0891  4             BEGIN
;  898          0892  4             IF NOT lib$cvt_dtb(.desc[dsc$w_length],
;  899          0893  4                                 .desc[dsc$a_pointer],
;  900          0894  4                                 window_value)
;  901          0895  4             THEN
;  902          0896  5                 BEGIN
;  903          0897  5                 SIGNAL(set$_syntax, 1, desc);
;  904          0898  5                 RETURN false;
;  905          0899  4                 END;
;  906          0900  4             IF .window_value LSS 7
;  907          0901  4             OR .window_value GTR 80
;  908          0902  4             THEN
;  909          0903  5                 BEGIN
;  910          0904  5                 SIGNAL(set$_syntax, 1, desc, set$_valerr);
;  911          0905  5                 RETURN false;
;  912          0906  4                 END;
;  913          0907  3             END;
;  914          0908  2         END;
;  915          0909  2
;  916          0910  2 RETURN true;
;  917          0911  1 END;



                                                                                  .PSECT  $PLIT$,NOWRT,NOEXE,2

                               44  45  53  53  45  43  43  41  00020 P.AAE:    .ASCII  \ACCESSED\                      ;
                                                     010E0008  00028 P.AAD:    .LONG   17694728                        ;
                                                     00000000' 0002C           .ADDRESS P.AAE                          ;
                               44  45  53  53  45  43  43  41  00030 P.AAG:    .ASCII  \ACCESSED\                      ;
                                                     010E0008  00038 P.AAF:    .LONG   17694728                        ;
                                                     00000000' 0003C           .ADDRESS P.AAG                          ;
           00  00  4B  43  45  48  43  5F  41  54  41  44  00040 P.AAI:    .ASCII  \DATA_CHECK\<0><0>              ;
                                                     010E000A  0004C P.AAH:    .LONG   17694730                        ;
                                                     00000000' 00050           .ADDRESS P.AAI                          ;
           00  00  4B  43  45  48  43  5F  41  54  41  44  00054 P.AAK:    .ASCII  \DATA_CHECK\<0><0>              ;
                                                     010E000A  00060 P.AAJ:    .LONG   17694730                        ;
                                                     00000000' 00064           .ADDRESS P.AAK                          ;
           00  00  4B  43  45  48  43  5F  41  54  41  44  00068 P.AAM:    .ASCII  \DATA_CHECK\<0><0>              ;
                                                     010E000A  00074 P.AAL:    .LONG   17694730                        ;
                                                     00000000' 00078           .ADDRESS P.AAM                          ;
                                           45  54  49  52  57  0007C P.AAN:    .ASCII  \WRITE\                         ;
                                                               00081           .BLKB   3
                                               44  41  45  52  00084 P.AAO:    .ASCII  \READ\                          ;
                                   45  54  49  52  57  4F  4E  00088 P.AAP:    .ASCII  \NOWRITE\                       ;
                                                               0008F           .BLKB   1
                                       44  41  45  52  4F  4E  00090 P.AAQ:    .ASCII  \NOREAD\                        ;
                                                               00096           .BLKB   2
45  54  45  4C  45  44  5F  4E  4F  5F  45  53  41  52  45  00098 P.AAS:    .ASCII  \ERASE_ON_DELETE\<0>            ;
                                                           00  000A7                                                   ;
                                                     010E000F  000A8 P.AAR:    .LONG   17694735                        ;
                                                     00000000' 000AC           .ADDRESS P.AAS                          ;
           00  00  00  4E  4F  49  53  4E  45  54  58  45  000B0 P.AAU:    .ASCII  \EXTENSION\<0><0><0>            ;
```

```
                                                  010E0009  000BC P.AAT:  .LONG   17694729
                                                  00000000' 000C0         .ADDRESS P.AAU
            00  00  00  4E  4F  49  53  4E  45  54  58  45  000C4 P.AAW:  .ASCII  \EXTENSION\<0><0><0>
                                                  010E0009  000D0 P.AAV:  .LONG   17694729
                                                  00000000' 000D4         .ADDRESS P.AAW
4E  4F  49  54  43  45  54  4F  52  50  5F  45  4C  49  46  000D8 P.AAY:  .ASCII  \FILE_PROTECTION\<0>
                                                        00  000E7
                                                  010E000F  000E8 P.AAX:  .LONG   17694735
                                                  00000000' 000EC         .ADDRESS P.AAY
4E  4F  49  54  43  45  54  4F  52  50  5F  45  4C  49  46  000F0 P.ABA:  .ASCII  \FILE_PROTECTION.SYSTEM\<0><0>
                        00  00  4D  45  54  53  59  53  2E  000FF
                                                  010E0016  00108 P.AAZ:  .LONG   17694742
                                                  00000000' 0010C         .ADDRESS P.ABA
4E  4F  49  54  43  45  54  4F  52  50  5F  45  4C  49  46  00110 P.ABC:  .ASCII  \FILE_PROTECTION.SYSTEM\<0><0>
                        00  00  4D  45  54  53  59  53  2E  0011F
                                                  010E0016  00128 P.ABB:  .LONG   17694742
                                                  00000000' 0012C         .ADDRESS P.ABC
4E  4F  49  54  43  45  54  4F  52  50  5F  45  4C  49  46  00130 P.ABE:  .ASCII  \FILE_PROTECTION.OWNER\<0><0><0>
                        00  00  00  52  45  4E  57  4F  2E  0013F
                                                  010E0015  00148 P.ABD:  .LONG   17694741
                                                  00000000' 0014C         .ADDRESS P.ABE
4E  4F  49  54  43  45  54  4F  52  50  5F  45  4C  49  46  00150 P.ABG:  .ASCII  \FILE_PROTECTION.OWNER\<0><0><0>
                        00  00  00  52  45  4E  57  4F  2E  0015F
                                                  010E0015  00168 P.ABF:  .LONG   17694741
                                                  00000000' 0016C         .ADDRESS P.ABG
4E  4F  49  54  43  45  54  4F  52  50  5F  45  4C  49  46  00170 P.ABI:  .ASCII  \FILE_PROTECTION.GROUP\<0><0><0>
                        00  00  00  50  55  4F  52  47  2E  0017F
                                                  010E0015  00188 P.ABH:  .LONG   17694741
                                                  00000000' 0018C         .ADDRESS P.ABI
4E  4F  49  54  43  45  54  4F  52  50  5F  45  4C  49  46  00190 P.ABK:  .ASCII  \FILE_PROTECTION.GROUP\<0><0><0>
                        00  00  00  50  55  4F  52  47  2E  0019F
                                                  010E0015  001A8 P.ABJ:  .LONG   17694741
                                                  00000000' 001AC         .ADDRESS P.ABK
4E  4F  49  54  43  45  54  4F  52  50  5F  45  4C  49  46  001B0 P.ABM:  .ASCII  \FILE_PROTECTION.WORLD\<0><0><0>
                        00  00  00  44  4C  52  4F  57  2E  001BF
                                                  010E0015  001C8 P.ABL:  .LONG   17694741
                                                  00000000' 001CC         .ADDRESS P.ABM
4E  4F  49  54  43  45  54  4F  52  50  5F  45  4C  49  46  001D0 P.ABO:  .ASCII  \FILE_PROTECTION.WORLD\<0><0><0>
                        00  00  00  44  4C  52  4F  57  2E  001DF
                                                  010E0015  001E8 P.ABN:  .LONG   17694741
                                                  00000000' 001EC         .ADDRESS P.ABO
49  4B  52  41  4D  5F  52  45  54  41  57  48  47  49  48  001F0 P.ABQ:  .ASCII  \HIGHWATER_MARKING\<0><0><0>
                                        00  00  00  47  4E  001FF
                                                  010E0011  00204 P.ABP:  .LONG   17694737
                                                  00000000' 00208         .ADDRESS P.ABQ
                            00  00  00  4C  45  42  41  4C  0020C P.ABS:  .ASCII  \LABEL\<0><0><0>
                                                  010E0005  00214 P.ABR:  .LONG   17694725
                                                  00000000' 00218         .ADDRESS P.ABS
                            00  00  00  4C  45  42  41  4C  0021C P.ABU:  .ASCII  \LABEL\<0><0><0>
                                                  010E0005  00224 P.ABT:  .LONG   17694725
                                                  00000000' 00228         .ADDRESS P.ABU
                                            00  47  4F  4C  0022C P.ABW:  .ASCII  \LOG\<0>
                                                  010E0003  00230 P.ABV:  .LONG   17694723
                                                  00000000' 00234         .ADDRESS P.ABW
54  41  43  49  46  49  52  45  56  5F  54  4E  55  4F  4D  00238 P.ABY:  .ASCII  \MOUNT_VERIFICATION\<0><0>
                                        00  00  4E  4F  49  00247
                                                  010E0012  0024C P.ABX:  .LONG   17694738
```

```
                                                 00000000'  00250           .ADDRESS P.ABY
               00  00  00  43  49  55  5F  52  45  4E  57  4F  00254 P.ACA:  .ASCII  \OWNER_UIC\<0><0><0>
                                                 010E0009  00260 P.ABZ:  .LONG   17694729
                                                 00000000'  00264           .ADDRESS P.ACA
               00  00  00  43  49  55  5F  52  45  4E  57  4F  00268 P.ACC:  .ASCII  \OWNER_UIC\<0><0><0>
                                                 010E0009  00274 P.ACB:  .LONG   17694729
                                                 00000000'  00278           .ADDRESS P.ACC
                                              0304  0004  0027C P.ACD:  .WORD   4, 772
                                                 00000000G  00280           .ADDRESS UIC_VALUE
                                      00000000  00000000  00284           .LONG   0, 0
               00  00  4E  4F  49  54  43  45  54  4F  52  50  0028C P.ACF:  .ASCII  \PROTECTION\<0><0>
                                                 010E000A  00298 P.ACE:  .LONG   17694730
                                                 00000000'  0029C           .ADDRESS P.ACF
54  53  59  53  2E  4E  4F  49  54  43  45  54  4F  52  50  002A0 P.ACH:  .ASCII  \PROTECTION.SYSTEM\<0><0><0>
                                      00  00  00  4D  45  002AF
                                                 010E0011  002B4 P.ACG:  .LONG   17694737
                                                 00000000'  002B8           .ADDRESS P.ACH
54  53  59  53  2E  4E  4F  49  54  43  45  54  4F  52  50  002BC P.ACJ:  .ASCII  \PROTECTION.SYSTEM\<0><0><0>
                                      00  00  00  4D  45  002CB
                                                 010E0011  002D0 P.ACI:  .LONG   17694737
                                                 00000000'  002D4           .ADDRESS P.ACJ
45  4E  57  4F  2E  4E  4F  49  54  43  45  54  4F  52  50  002D8 P.ACL:  .ASCII  \PROTECTION.OWNER\
                                                      52  002E7
                                                 010E0010  002E8 P.ACK:  .LONG   17694736
                                                 00000000'  002EC           .ADDRESS P.ACL
45  4E  57  4F  2E  4E  4F  49  54  43  45  54  4F  52  50  002F0 P.ACN:  .ASCII  \PROTECTION.OWNER\
                                                      52  002FF
                                                 010E0010  00300 P.ACM:  .LONG   17694736
                                                 00000000'  00304           .ADDRESS P.ACN
55  4F  52  47  2E  4E  4F  49  54  43  45  54  4F  52  50  00308 P.ACP:  .ASCII  \PROTECTION.GROUP\
                                                      50  00317
                                                 010E0010  00318 P.ACO:  .LONG   17694736
                                                 00000000'  0031C           .ADDRESS P.ACP
55  4F  52  47  2E  4E  4F  49  54  43  45  54  4F  52  50  00320 P.ACR:  .ASCII  \PROTECTION.GROUP\
                                                      50  0032F
                                                 010E0010  00330 P.ACQ:  .LONG   17694736
                                                 00000000'  00334           .ADDRESS P.ACR
4C  52  4F  57  2E  4E  4F  49  54  43  45  54  4F  52  50  00338 P.ACT:  .ASCII  \PROTECTION.WORLD\
                                                      44  00347
                                                 010E0010  00348 P.ACS:  .LONG   17694736
                                                 00000000'  0034C           .ADDRESS P.ACT
4C  52  4F  57  2E  4E  4F  49  54  43  45  54  4F  52  50  00350 P.ACV:  .ASCII  \PROTECTION.WORLD\
                                                      44  0035F
                                                 010E0010  00360 P.ACU:  .LONG   17694736
                                                 00000000'  00364           .ADDRESS P.ACV
                           00  44  4C  49  55  42  45  52  00368 P.ACX:  .ASCII  \REBUILD\<0>
                                                 010E0007  00370 P.ACW:  .LONG   17694727
                                                 00000000'  00374           .ADDRESS P.ACX
               00  00  00  4E  4F  49  54  4E  45  54  45  52  00378 P.ACZ:  .ASCII  \RETENTION\<0><0><0>
                                                 010E0009  00384 P.ACY:  .LONG   17694729
                                                 00000000'  00388           .ADDRESS P.ACZ
               00  00  00  4E  4F  49  54  4E  45  54  45  52  0038C P.ADB:  .ASCII  \RETENTION\<0><0><0>
                                                 010E0009  00398 P.ADA:  .LONG   17694729
                                                 00000000'  0039C           .ADDRESS P.ADB
               00  00  00  4E  4F  49  54  4E  45  54  45  52  003A0 P.ADD:  .ASCII  \RETENTION\<0><0><0>
                                                 010E0009  003AC P.ADC:  .LONG   17694729
                                                 00000000'  003B0           .ADDRESS P.ADD
```

```
                    00  00  44  41  4F  4C  4E  55  003B4 P.ADF:    .ASCII  \UNLOAD\<0><0>                                   ;
                                          010E0006  003BC P.ADE:    .LONG   17694726                                         ;
                                          00000000' 003C0           .ADDRESS P.ADF                                           ;
 00  00  00  45  4D  41  4E  5F  52  45  53  55  003C4 P.ADH:    .ASCII  \USER_NAME\<0><0><0>                             ;
                                          010E0009  003D0 P.ADG:    .LONG   17694729                                         ;
                                          00000000' 003D4           .ADDRESS P.ADH                                           ;
 00  00  00  45  4D  41  4E  5F  52  45  53  55  003D8 P.ADJ:    .ASCII  \USER_NAME\<0><0><0>                             ;
                                          010E0009  003E4 P.ADI:    .LONG   17694729                                         ;
                                          00000000' 003E8           .ADDRESS P.ADJ                                           ;
                    00  53  57  4F  44  4E  49  57  003EC P.ADL:    .ASCII  \WINDOWS\<0>                                     ;
                                          010E0007  003F4 P.ADK:    .LONG   17694727                                         ;
                                          00000000' 003F8           .ADDRESS P.ADL                                           ;
                    00  53  57  4F  44  4E  49  57  003FC P.ADN:    .ASCII  \WINDOWS\<0>                                     ;
                                          010E0007  00404 P.ADM:    .LONG   17694727                                         ;
                                          00000000' 00408           .ADDRESS P.ADN                                           ;

                                                          SETPRO_MASK=          FPROT_VALUE+2
                                                          SETPRO_PROT=          FPROT_VALUE
                                                          SETPRO_MASK=          VPROT_VALUE+2
                                                          SETPRO_PROT=          VPROT_VALUE
                                                                    .EXTRN  SYS$SETPRV, SYS$GETJPIW

                                                                    .PSECT  $CODE$,NOWRT,2

                                              0FFC 00000 GET_QUALS:
                                                                    .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11            ; 0431
                     5B 00000000G  00  9E 00002           MOVAB   CLI$GET_VALUE, R11                                ;
                     5A 00000000G  00  9E 00009           MOVAB   CLI$PRESENT, R10                                  ;
                     59 00000000'  EF  9E 00010           MOVAB   FLAGS, R9                                         ;
                     58 00000000'  EF  9E 00017           MOVAB   RETMIN_VALUE, R8                                  ;
                     57 00000000'  EF  9E 0001E           MOVAB   P.AAD, R7                                         ;
                     5E            28  C2 00025           SUBL2   #40, SP                                           ;
              20     AE 020E0000   8F  D0 00028           MOVL    #34471936, DESC                                   ; 0448
                             24    AE  D4 00030           CLRL    DESC+4                                            ;
                                   57  DD 00033           PUSHL   R7                                                ; 0453
                     6A            01  FB 00035           CALLS   #1, CLI$PRESENT                                   ;
                     62            50  E9 00038           BLBC    R0, 7$                                            ;
                     69            02  88 0003B           BISB2   #2, FLAGS                                         ; 0457
                             18    AE  9F 0003E           PUSHAB  PRIVS                                             ; 0466
                                   01  DD 00041           PUSHL   #1                                                ;
                     7E            01  7D 00043           MOVQ    #1, -(SP)                                         ;
         00000000G   00            04  FB 00046           CALLS   #4, SYS$SETPRV                                    ;
                     56            50  D0 0004D           MOVL    R0, STATUS                                        ;
                     03            56  E8 00050           BLBS    STATUS, 1$                                        ;
                                 04C9  31 00053           BRW     48$                                               ;
    09         1A    AE            02  E0 00056 1$:       BBS     #2, PRIVS+2, 2$                                   ; 0472
                        00000000G  8F  DD 0005B           PUSHL   #SET$_OPERREQ                                     ; 0475
                                 04BD  31 00061           BRW     49$                                               ;
               EC    A8            03  D0 00064 2$:       MOVL    #3, ACC_VALUE                                     ; 0482
                             20    AE  9F 00068           PUSHAB  DESC                                              ; 0487
                             10    A7  9F 0006B           PUSHAB  P.AAF                                             ;
                     6B            02  FB 0006E           CALLS   #2, CLI$GET_VALUE                                 ;
                     29            50  E9 00071           BLBC    R0, 7$                                            ;
                             EC    A8  9F 00074           PUSHAB  ACC_VALUE                                         ; 0490
                             28    AE  DD 00077           PUSHL   DESC+4                                            ; 0491
                     7E      28    AE  3C 0007A           MOVZWL  DESC, -(SP)                                       ; 0490
         00000000G   00            03  FB 0007E           CALLS   #3, LIB$CVT_DTB                                   ;
```

```
                              03              50  E8 00085          BLBS    R0, 4$                          ;
                                            04F4  31 00088 3$:      BRW     53$                             ;
                              50        EC    A8  D0 0008B 4$:      MOVL    ACC_VALUE, R0                   ; 0498
                                              03  18 0008F          BGEQ    6$                              ;
                                            050F  31 00091 5$:      BRW     57$                             ;
                   000000FF   8F              50  D1 00094 6$:      CMPL    R0, #255                        ; 0499
                                              F4  14 0009B          BGTR    5$                              ;
                                        24    A7  9F 0009D 7$:      PUSHAB  P.AAH                           ; 0511
                              6A              01  FB 000A0          CALLS   #1, CLI$PRESENT                 ;
                              5D              50  E9 000A3          BLBC    R0, 12$                         ;
                              69              04  88 000A6          BISB2   #4, FLAGS                       ; 0514
                                        20    AE  9F 000A9          PUSHAB  DESC                            ; 0515
                                        38    A7  9F 000AC          PUSHAB  P.AAJ                           ;
                              6B              02  FB 000AF          CALLS   #2, CLI$GET_VALUE               ;
                              06              50  E8 000B2          BLBS    R0, 8$                          ;
                         04   A9              04  88 000B5          BISB2   #4, DFLAGS                      ; 0517
                                              48  11 000B9          BRB     12$                             ;
                                        20    AE  9F 000BB 8$:      PUSHAB  DESC                            ; 0519
                                        4C    A7  9F 000BE          PUSHAB  P.AAL                           ;
                              6B              02  FB 000C1          CALLS   #2, CLI$GET_VALUE               ;
                              3C              50  E9 000C4          BLBC    R0, 12$                         ;
                              54        20    AE  3C 000C7          MOVZWL  DESC, R4                        ; 0521
           54     A7     24   BE              54  29 000CB          CMPC3   R4, @DESC+4, P.AAN              ;
                                              06  12 000D1          BNEQ    9$                              ;
                         04   A9              04  88 000D3          BISB2   #4, DFLAGS                      ; 0523
                                              E2  11 000D7          BRB     8$                              ;
           5C     A7     24   BE              54  29 000D9 9$:      CMPC3   R4, @DESC+4, P.AAO              ; 0524
                                              06  12 000DF          BNEQ    10$                             ;
                         04   A9              02  88 000E1          BISB2   #2, DFLAGS                      ; 0526
                                              D4  11 000E5          BRB     8$                              ;
           60     A7     24   BE              54  29 000E7 10$:     CMPC3   R4, @DESC+4, P.AAP              ; 0527
                                              06  12 000ED          BNEQ    11$                             ;
                         04   A9              10  88 000EF          BISB2   #16, DFLAGS                     ; 0529
                                              C6  11 000F3          BRB     8$                              ;
           68     A7     24   BE              54  29 000F5 11$:     CMPC3   R4, @DESC+4, P.AAQ              ; 0530
                                              8B  12 000FB          BNEQ    3$                              ;
                         04   A9              08  88 000FD          BISB2   #8, DFLAGS                      ; 0532
                                              B8  11 00101          BRB     8$                              ;
                                      0080    C7  9F 00103 12$:     PUSHAB  P.AAR                           ; 0544
                              6A              01  FB 00107          CALLS   #1, CLI$PRESENT                 ;
                              56              50  D0 0010A          MOVL    R0, STATUS                      ;
                   00000000G  8F              56  D1 0010D          CMPL    STATUS, #CLI$_ABSENT            ; 0545
                                              0A  13 00114          BEQL    13$                             ;
                         01   A9              10  88 00116          BISB2   #16, FLAGS+1                    ; 0548
01     A9         01          05              56  F0 0011A          INSV    STATUS, #5, #1, FLAGS+1         ; 0549
                                      0094    C7  9F 00120 13$:     PUSHAB  P.AAT                           ; 0555
                              6A              01  FB 00124          CALLS   #1, CLI$PRESENT                 ;
                              46              50  E9 00127          BLBC    R0, 17$                         ;
                              69              08  88 0012A          BISB2   #8, FLAGS                       ; 0558
                   00000000G  00              05  D0 0012D          MOVL    #5, EXTE_VALUE                  ; 0559
                                        20    AE  9F 00134          PUSHAB  DESC                            ; 0560
                                      00A8    C7  9F 00137          PUSHAB  P.AAV                           ;
                              6B              02  FB 0013B          CALLS   #2, CLI$GET_VALUE               ;
                              2F              50  E9 0013E          BLBC    R0, 17$                         ;
                                 00000000G    00  9F 00141          PUSHAB  EXTE_VALUE                      ; 0563
                                        28    AE  DD 00147          PUSHL   DESC+4                          ; 0564
                              7E        28    AE  3C 0014A          MOVZWL  DESC, -(SP)                     ; 0563
```

```
         00000000G  00              03  FB 0014E         CALLS   #3, LIB$CVT_DTB                      ;
                    03              50  E8 00155         BLBS    R0, 14$                              ;
                                  0424  31 00158         BRW     53$                                  ;
                 50 00000000G       00  D0 0015B 14$:    MOVL    EXTE_VALUE, R0                       ; 0571
                                    03  18 00162         BGEQ    16$                                  ;
                                  043C  31 00164 15$:    BRW     57$                                  ;
         0000FFFF   8F              50  D1 00167 16$:    CMPL    R0, #65535                           ; 0572
                                    F4  14 0016E         BGTR    15$                                  ;
                            00C0    C7  9F 00170 17$:    PUSHAB  P.AAX                                ; 0584
                    6A              01  FB 00174         CALLS   #1, CLI$PRESENT                      ;
                    03              50  E8 00177         BLBS    R0, 18$                              ;
                                  00B7  31 0017A         BRW     22$                                  ;
                    69              10  88 0017D 18$:    BISB2   #16, FLAGS                           ; 0591
                              F0    A8  D4 00180         CLRL    FPROT_VALUE                          ; 0592
                            00E0    C7  9F 00183         PUSHAB  P.AAZ                                ; 0594
                    6A              01  FB 00187         CALLS   #1, CLI$PRESENT                      ;
                    1F              50  E9 0018A         BLBC    R0, 19$                              ;
               F2   A8              0F  88 0018D         BISB2   #15, SETPRO_MASK                     ; 0597
                              20    AE  9F 00191         PUSHAB  DESC                                 ; 0598
                            0100    C7  9F 00194         PUSHAB  P.ABB                                ;
                    6B              02  FB 00198         CALLS   #2, CLI$GET_VALUE                    ;
                    0E              50  E9 0019B         BLBC    R0, 19$                              ;
                              20    AE  9F 0019E         PUSHAB  DESC                                 ; 0599
         00000000V  EF              01  FB 001A1         CALLS   #1, PARSE_CLASS                      ;
               F0   A8              50  B0 001A8         MOVW    R0, SETPRO_PROT                      ;
                            0120    C7  9F 001AC 19$:    PUSHAB  P.ABD                                ; 0601
                    6A              01  FB 001B0         CALLS   #1, CLI$PRESENT                      ;
                    23              50  E9 001B3         BLBC    R0, 20$                              ;
               F2   A8        F0    8F  88 001B6         BISB2   #240, SETPRO_MASK                    ; 0604
                              20    AE  9F 001BB         PUSHAB  DESC                                 ; 0605
                            0140    C7  9F 001BE         PUSHAB  P.ABF                                ;
                    6B              02  FB 001C2         CALLS   #2, CLI$GET_VALUE                    ;
                    11              50  E9 001C5         BLBC    R0, 20$                              ;
                              20    AE  9F 001C8         PUSHAB  DESC                                 ; 0606
         00000000V  EF              01  FB 001CB         CALLS   #1, PARSE_CLASS                      ;
                    50              10  C4 001D2         MULL2   #16, R0                              ;
               F0   A8              50  A8 001D5         BISW2   R0, SETPRO_PROT                      ;
                            0160    C7  9F 001D9 20$:    PUSHAB  P.ABH                                ; 0608
                    6A              01  FB 001DD         CALLS   #1, CLI$PRESENT                      ;
                    23              50  E9 001E0         BLBC    R0, 21$                              ;
               F3   A8              0F  88 001E3         BISB2   #15, SETPRO_MASK+1                   ; 0611
                              20    AE  9F 001E7         PUSHAB  DESC                                 ; 0612
                            0180    C7  9F 001EA         PUSHAB  P.ABJ                                ;
                    6B              02  FB 001EE         CALLS   #2, CLI$GET_VALUE                    ;
                    12              50  E9 001F1         BLBC    R0, 21$                              ;
                              20    AE  9F 001F4         PUSHAB  DESC                                 ; 0613
         00000000V  EF              01  FB 001F7         CALLS   #1, PARSE_CLASS                      ;
      50            50              08  78 001FE         ASHL    #8, R0, R0                           ;
               F0   A8              50  A8 00202         BISW2   R0, SETPRO_PROT                      ;
                            01A0    C7  9F 00206 21$:    PUSHAB  P.ABL                                ; 0615
                    6A              01  FB 0020A         CALLS   #1, CLI$PRESENT                      ;
                    24              50  E9 0020D         BLBC    R0, 22$                              ;
               F3   A8        F0    8F  88 00210         BISB2   #240, SETPRO_MASK+1                  ; 0618
                              20    AE  9F 00215         PUSHAB  DESC                                 ; 0619
                            01C0    C7  9F 00218         PUSHAB  P.ABN                                ;
                    6B              02  FB 0021C         CALLS   #2, CLI$GET_VALUE                    ;
                    12              50  E9 0021F         BLBC    R0, 22$                              ;
```

```
                                          20   AE   9F 00222        PUSHAB  DESC                                      ; 0620
                       00000000V   EF           01   FB 00225        CALLS   #1, PARSE_CLASS                           ;
                  50               50           0C   78 0022C        ASHL    #12, R0, R0                               ;
                             F0    A8           50   A8 00230        BISW2   R0, SETPRO_PROT                           ;
                                        01DC    C7   9F 00234 22$:   PUSHAB  P.ABP                                     ; 0627
                                   6A           01   FB 00238        CALLS   #1, CLI$PRESENT                           ;
                                   56           50   D0 0023B        MOVL    R0, STATUS                                ;
                       00000000G   8F           56   D1 0023E        CMPL    STATUS, #CLI$_ABSENT                      ; 0628
                                   0E                13 00245        BEQL    23$                                       ;
                             01    A9     40    8F   88 00247        BISB2   #64, FLAGS+1                              ; 0631
                                   50           56   D2 0024C        MCOML   STATUS, R0                                ; 0632
      01    A9            01       07           50   F0 0024F        INSV    R0, #7, #1, FLAGS+1                       ;
                                        01EC    C7   9F 00255 23$:   PUSHAB  P.ABR                                     ; 0638
                                   6A           01   FB 00259        CALLS   #1, CLI$PRESENT                           ;
                                   2E           50   E9 0025C        BLBC    R0, 25$                                   ;
                                          20    AE   9F 0025F        PUSHAB  DESC                                      ; 0640
                                        01FC    C7   9F 00262        PUSHAB  P.ABT                                     ;
                                   6B           02   FB 00266        CALLS   #2, CLI$GET_VALUE                         ;
                                   21           50   E9 00269        BLBC    R0, 25$                                   ;
                                   69           20   88 0026C        BISB2   #32, FLAGS                                ; 0643
                                   0C     20    AE   B1 0026F        CMPW    DESC, #12                                 ; 0644
                                   03                1B 00273        BLEQU   24$                                       ;
                                        0307         31 00275        BRW     53$                                       ;
                             F4    A8     20    AE   3C 00278 24$:   MOVZWL  DESC, LABEL_VALUE                         ; 0650
                             F8    A8     24    AE   D0 0027D        MOVL    DESC+4, LABEL_VALUE+4                     ; 0651
                             20    AE 020E0000  8F   D0 00282        MOVL    #34471936, DESC                           ; 0652
                                          24    AE   D4 0028A        CLRL    DESC+4                                    ;
                                        0208    C7   9F 0028D 25$:   PUSHAB  P.ABV                                     ; 0658
                                   6A           01   FB 00291        CALLS   #1, CLI$PRESENT                           ;
            69            01       06           50   F0 00294        INSV    R0, #6, #1, FLAGS                         ;
                                        0224    C7   9F 00299        PUSHAB  P.ABX                                     ; 0663
                                   6A           01   FB 0029D        CALLS   #1, CLI$PRESENT                           ;
                                   56           50   D0 002A0        MOVL    R0, STATUS                                ;
                       00000000G   8F           56   D1 002A3        CMPL    STATUS, #CLI$_ABSENT                      ; 0664
                                   0A                13 002AA        BEQL    26$                                       ;
                             02    A9           01   88 002AC        BISB2   #1, FLAGS+2                               ; 0667
      02    A9            01       01           56   F0 002B0        INSV    STATUS, #1, #1, FLAGS+2                   ; 0668
                                        0238    C7   9F 002B6 26$:   PUSHAB  P.ABZ                                     ; 0674
                                   6A           01   FB 002BA        CALLS   #1, CLI$PRESENT                           ;
                                   45           50   E9 002BD        BLBC    R0, 29$                                   ;
                                   69     80    8F   88 002C0        BISB2   #128, FLAGS                               ; 0677
                                          20    AE   9F 002C4        PUSHAB  DESC                                      ; 0678
                                        024C    C7   9F 002C7        PUSHAB  P.ACB                                     ;
                                   6B           02   FB 002CB        CALLS   #2, CLI$GET_VALUE                         ;
                                   24           50   E8 002CE        BLBS    R0, 28$                                   ;
                                                7E   7C 002D1        CLRQ    -(SP)                                     ; 0687
                                          20    AE   9F 002D3        PUSHAB  IOSB                                      ;
                                        0254    C7   9F 002D6        PUSHAB  P.ACD                                     ;
                                                7E   7C 002DA        CLRQ    -(SP)                                     ;
                                                7E   D4 002DC        CLRL    -(SP)                                     ;
                       00000000G   00           07   FB 002DE        CALLS   #7, SYS$GETJPIW                           ;
                                   56           50   D0 002E5        MOVL    R0, STATUS                                ;
                                   07           56   E9 002E8        BLBC    STATUS, 27$                               ; 0688
                                   56     18    AE   3C 002EB        MOVZWL  IOSB, STATUS                              ; 0689
                                   13           56   E8 002EF        BLBS    STATUS, 29$                               ; 0690
                                        022A         31 002F2 27$:   BRW     48$                                       ; 0693
                                   00000000G    00   9F 002F5 28$:   PUSHAB  UIC_VALUE                                 ; 0697
```

```
                                      24    AE   9F  002FB         PUSHAB  DESC                                      :
                       00000000G  00        02   FB  002FE         CALLS   #2, PARSE_UIC                             :
                                    0270    C7   9F  00305  29$:   PUSHAB  P.ACE                                     : 0703
                                  6A        01   FB  00309         CALLS   #1, CLI$PRESENT                           :
                                  03        50   E8  0030C         BLBS    R0, 30$                                   :
                                      00B8       31  0030F         BRW     34$                                       :
                             01   A9        04   88  00312  30$:   BISB2   #4, FLAGS+1                               : 0710
                                      FC    A8   D4  00316         CLRL    VPROT_VALUE                               : 0711
                                    028C    C7   9F  00319         PUSHAB  P.ACG                                     : 0713
                                  6A        01   FB  0031D         CALLS   #1, CLI$PRESENT                           :
                                  1F        50   E9  00320         BLBC    R0, 31$                                   :
                             FE   A8        0F   88  00323         BISB2   #15, SETPRO_MASK                          : 0716
                                      20    AE   9F  00327         PUSHAB  DESC                                      : 0717
                                    02A8    C7   9F  0032A         PUSHAB  P.ACI                                     :
                                  6B        02   FB  0032E         CALLS   #2, CLI$GET_VALUE                         :
                                  0E        50   E9  00331         BLBC    R0, 31$                                   :
                                      20    AE   9F  00334         PUSHAB  DESC                                      : 0718
                       00000000V  EF        01   FB  00337         CALLS   #1, PARSE_CLASS                           :
                             FC   A8        50   B0  0033E         MOVW    R0, SETPRO_PROT                           :
                                    02C0    C7   9F  00342  31$:   PUSHAB  P.ACK                                     : 0720
                                  6A        01   FB  00346         CALLS   #1, CLI$PRESENT                           :
                                  23        50   E9  00349         BLBC    R0, 32$                                   :
                             FE   A8  F0    8F   88  0034C         BISB2   #240, SETPRO_MASK                         : 0723
                                      20    AE   9F  00351         PUSHAB  DESC                                      : 0724
                                    02D8    C7   9F  00354         PUSHAB  P.ACM                                     :
                                  6B        02   FB  00358         CALLS   #2, CLI$GET_VALUE                         :
                                  11        50   E9  0035B         BLBC    R0, 32$                                   :
                                      20    AE   9F  0035E         PUSHAB  DESC                                      : 0725
                       00000000V  EF        01   FB  00361         CALLS   #1, PARSE_CLASS                           :
                                  50        10   C4  00368         MULL2   #16, R0                                   :
                             FC   A8        50   A8  0036B         BISW2   R0, SETPRO_PROT                           :
                                    02F0    C7   9F  0036F  32$:   PUSHAB  P.ACO                                     : 0727
                                  6A        01   FB  00373         CALLS   #1, CLI$PRESENT                           :
                                  23        50   E9  00376         BLBC    R0, 33$                                   :
                             FF   A8        0F   88  00379         BISB2   #15, SETPRO_MASK+1                        : 0730
                                      20    AE   9F  0037D         PUSHAB  DESC                                      : 0731
                                    0308    C7   9F  00380         PUSHAB  P.ACQ                                     :
                                  6B        02   FB  00384         CALLS   #2, CLI$GET_VALUE                         :
                                  12        50   E9  00387         BLBC    R0, 33$                                   :
                                      20    AE   9F  0038A         PUSHAB  DESC                                      : 0732
                       00000000V  EF        01   FB  0038D         CALLS   #1, PARSE_CLASS                           :
                     50           50        08   78  00394         ASHL    #8, R0, R0                                :
                             FC   A8        50   A8  00398         BISW2   R0, SETPRO_PROT                           :
                                    0320    C7   9F  0039C  33$:   PUSHAB  P.ACS                                     : 0734
                                  6A        01   FB  003A0         CALLS   #1, CLI$PRESENT                           :
                                  24        50   E9  003A3         BLBC    R0, 34$                                   :
                             FF   A8  F0    8F   88  003A6         BISB2   #240, SETPRO_MASK+1                       : 0737
                                      20    AE   9F  003AB         PUSHAB  DESC                                      : 0738
                                    0338    C7   9F  003AE         PUSHAB  P.ACU                                     :
                                  6B        02   FB  003B2         CALLS   #2, CLI$GET_VALUE                         :
                                  12        50   E9  003B5         BLBC    R0, 34$                                   :
                                      20    AE   9F  003B8         PUSHAB  DESC                                      : 0739
                       00000000V  EF        01   FB  003BB         CALLS   #1, PARSE_CLASS                           :
                     50           50        0C   78  003C2         ASHL    #12, R0, R0                               :
                             FC   A8        50   A8  003C6         BISW2   R0, SETPRO_PROT                           :
                                    0348    C7   9F  003CA  34$:   PUSHAB  P.ACW                                     : 0746
                                  6A        01   FB  003CE         CALLS   #1, CLI$PRESENT                           :
```

```
                                                                  N  2
SETVOL                                                           16-Sep-1984 01:01:55    VAX-11 Bliss-32 V4.0-742                                   Page  31
V04-000                                                          14-Sep-1984 12:09:22    [CLIUTL.SRC]SETVOLUME.B32;1                                     (7)

                                        56            50  D0 003D1         MOVL    R0, STATUS                                   ;
                            00000000G   8F            56  D1 003D4         CMPL    STATUS, #CLI$_ABSENT                         ; 0747
                                                      0A  13 003DB         BEQL    35$                                          ;
                                   02   A9            10  88 003DD         BISB2   #16, FLAGS+2                                 ; 0750
      02    A9            01            05            56  F0 003E1         INSV    STATUS, #5, #1, FLAGS+2                      ; 0751
                                               035C   C7  9F 003E7 35$:    PUSHAB  P.ACY                                        ; 0757
                                        6A            01  FB 003EB         CALLS   #1, CLI$PRESENT                              ;
                                        69            50  E9 003EE         BLBC    R0, 40$                                      ;
                                   01   A9            01  88 003F1         BISB2   #1, FLAGS+1                                  ; 0762
            08            00            6E            00  2C 003F5         MOVC5   #0, (SP), #0, #8, RETMIN_VALUE               ; 0764
                                                      68     003FA                                                              ;
            08            00            6E            00  2C 003FB         MOVC5   #0, (SP), #0, #8, RETMAX_VALUE               ; 0765
                                                 08   A8     00400                                                              ;
                                                 20   AE  9F 00402         PUSHAB  DESC                                         ; 0770
                                               0370   C7  9F 00405         PUSHAB  P.ADA                                        ;
                                        6B            02  FB 00409         CALLS   #2, CLI$GET_VALUE                            ;
                                        03            50  E8 0040C         BLBS    R0, 36$                                      ;
                                               016D   31     0040F         BRW     53$                                          ;
                                                 18   AE  9F 00412 36$:    PUSHAB  TEMP_DESC                                    ; 0780
                                                 24   AE  9F 00415         PUSHAB  DESC                                         ;
                            00000000G   00            02  FB 00418         CALLS   #2, LIB$CVT_DTIME                            ;
                                        56            50  D0 0041F         MOVL    R0, STATUS                                   ;
                                        04            56  E8 00422         BLBS    STATUS, 37$                                  ;
                                                      58  DD 00425         PUSHL   R8                                           ; 0783
                                                      28  11 00427         BRB     38$                                          ;
                          68       18   AE            08  28 00429 37$:    MOVC3   #8, TEMP_DESC, RETMIN_VALUE                  ; 0786
                                                 20   AE  9F 0042E         PUSHAB  DESC                                         ; 0792
                                               0384   C7  9F 00431         PUSHAB  P.ADC                                        ;
                                        6B            02  FB 00435         CALLS   #2, CLI$GET_VALUE                            ;
                                        21            50  E9 00438         BLBC    R0, 41$                                      ;
                                                 18   AE  9F 0043B         PUSHAB  TEMP_DESC                                    ; 0795
                                                 24   AE  9F 0043E         PUSHAB  DESC                                         ;
                            00000000G   00            02  FB 00441         CALLS   #2, LIB$CVT_DTIME                            ;
                                        56            50  D0 00448         MOVL    R0, STATUS                                   ;
                                        06            56  E8 0044B         BLBS    STATUS, 39$                                  ;
                                                 08   A8  9F 0044E         PUSHAB  RETMAX_VALUE                                 ; 0798
                                               012E   31  00451 38$:       BRW     54$                                          ;
                     08   A8       18   AE            08  28 00454 39$:    MOVC3   #8, TEMP_DESC, RETMAX_VALUE                  ; 0801
                                                      56  11 0045A 40$:    BRB     46$                                          ; 0792
                                        6E D71BC000   8F  D0 0045C 41$:    MOVL    #-686047232, ONE_WEEK                        ; 0811
                                   04   AE     FA7F   8F  32 00463         CVTWL   #-1409, ONE_WEEK+4                           ;
                     10   AE            68            68  C1 00469         ADDL3   RETMIN_VALUE, RETMIN_VALUE, DOUBLE           ; 0818
                                   14   AE       04   A8  D0 0046E         MOVL    RETMIN_VALUE+4, DOUBLE                       ;
                                   14   AE       14   AE  D8 00473         ADWC    DOUBLE, DOUBLE                               ;
                     08   AE            68            6E  C1 00478         ADDL3   ONE_WEEK, RETMIN_VALUE, WEEK_PLUS            ; 0819
                                   0C   AE       04   A8  D0 0047D         MOVL    RETMIN_VALUE+4, WEEK_PLUS                    ;
                                   0C   AE       04   AE  D8 00482         ADWC    ONE_WEEK, WEEK_PLUS                          ;
                                        50            01  CE 00487         MNEGL   #1, R0                                       ; 0820
                                   0C   AE       14   AE  D1 0048A         CMPL    DOUBLE, WEEK_PLUS                            ;
                                                      0F  19 0048F         BLSS    44$                                          ;
                                                      09  14 00491         BGTR    42$                                          ;
                                   08   AE       10   AE  D1 00493         CMPL    DOUBLE, WEEK_PLUS                            ;
                                                      04  13 00498         BEQL    43$                                          ;
                                                      04  1F 0049A         BLSSU   44$                                          ;
                                                      50  D6 0049C 42$:    INCL    R0                                           ;
                                                      50  D6 0049E 43$:    INCL    R0                                           ;
                                                      50  D5 004A0 44$:    TSTL    R0                                           ;
```

```
                                                  08  15 004A2        BLEQ    45$                                 ;
                   08   A8         10 AE          08  28 004A4        MOVC3   #8, DOUBLE, RETMAX_VALUE            ; 0821
                                                  06  11 004AA        BRB     46$                                 ;
                   08   A8         08 AE          08  28 004AC 45$:   MOVC3   #8, WEEK_PLUS, RETMAX_VALUE         ; 0822
                                             0394 C7  9F 004B2 46$:   PUSHAB  P.ADE                               ; 0829
                                      6A          01  FB 004B6        CALLS   #1, CLI$PRESENT                     ;
                                      56          50  D0 004B9        MOVL    R0, STATUS                          ;
                            00000000G 8F          56  D1 004BC        CMPL    STATUS, #CLI$_ABSENT                ; 0830
                                                  0A  13 004C3        BEQL    47$                                 ;
                                   02 A9          04  88 004C5        BISB2   #4, FLAGS+2                         ; 0833
        02   A9         01            03          56  F0 004C9        INSV    STATUS, #3, #1, FLAGS+2             ; 0834
                                             03A8 C7  9F 004CF 47$:   PUSHAB  P.ADG                               ; 0840
                                      6A          01  FB 004D3        CALLS   #1, CLI$PRESENT                     ;
                                      73          50  E9 004D6        BLBC    R0, 52$                             ;
                                   01 A9          02  88 004D9        BISB2   #2, FLAGS+1                         ; 0843
                                               20 AE  9F 004DD        PUSHAB  DESC                                ; 0844
                                             03BC C7  9F 004E0        PUSHAB  P.ADI                               ;
                                      6B          02  FB 004E4        CALLS   #2, CLI$GET_VALUE                   ;
                                      47          50  E8 004E7        BLBS    R0, 51$                             ;
                                      50       10 AE  9E 004EA        MOVAB   JPI_LIST, $$ITMBLKPTR               ; 0854
                                      80 0202000C 8F  D0 004EE        MOVL    #33685516, ($$ITMBLKPTR)+           ;
                                      80       08 A9  9E 004F5        MOVAB   USER_LABEL, ($$ITMBLKPTR)+          ;
                                      80       10 A8  9E 004F9        MOVAB   USER_VALUE, ($$ITMBLKPTR)+          ;
                                                  80  D4 004FD        CLRL    ($$ITMBLKPTR)+                      ;
                                                  7E  7C 004FF        CLRQ    -(SP)                               ; 0856
                                               10 AE  9F 00501        PUSHAB  IOSB                                ;
                                               1C AE  9F 00504        PUSHAB  JPI_LIST                            ;
                                                  7E  7C 00507        CLRQ    -(SP)                               ;
                                                  7E  D4 00509        CLRL    -(SP)                               ;
                            00000000G 00          07  FB 0050B        CALLS   #7, SYS$GETJPIW                     ;
                                      56          50  D0 00512        MOVL    R0, STATUS                          ;
                                      07          56  E9 00515        BLBC    STATUS, 48$                         ; 0857
                                      56       08 AE  3C 00518        MOVZWL  IOSB, STATUS                        ; 0858
                                      0B          56  E8 0051C        BLBS    STATUS, 50$                         ; 0859
                                                  56  DD 0051F 48$:   PUSHL   STATUS                              ; 0862
                            00000000G 00          01  FB 00521 49$:   CALLS   #1, LIB$SIGNAL                      ;
                                                  67  11 00528        BRB     55$                                 ; 0863
                                   14 A8       08 A9  9E 0052A 50$:   MOVAB   USER_LABEL, USER_VALUE+4            ; 0865
                                                  1B  11 0052F        BRB     52$                                 ; 0844
                                   0C          20 AE  B1 00531 51$:   CMPW    DESC, #12                           ; 0869
                                                  48  1A 00535        BGTRU   53$                                 ;
                                   10 A8       20 AE  3C 00537        MOVZWL  DESC, USER_VALUE                    ; 0875
                                   14 A8       24 AE  D0 0053C        MOVL    DESC+4, USER_VALUE+4                ; 0876
                                   20 AE 020E0000 8F  D0 00541        MOVL    #34471936, DESC                     ; 0877
                                               24 AE  D4 00549        CLRL    DESC+4                              ;
                                             03CC C7  9F 0054C 52$:   PUSHAB  P.ADK                               ; 0884
                                      6A          01  FB 00550        CALLS   #1, CLI$PRESENT                     ;
                                      67          50  E9 00553        BLBC    R0, 58$                             ;
                                   01 A9          08  88 00556        BISB2   #8, FLAGS+1                         ; 0887
                                   18 A8          07  D0 0055A        MOVL    #7, WINDOW_VALUE                    ; 0888
                                               20 AE  9F 0055E        PUSHAB  DESC                                ; 0889
                                             03DC C7  9F 00561        PUSHAB  P.ADM                               ;
                                      6B          02  FB 00565        CALLS   #2, CLI$GET_VALUE                   ;
                                      52          50  E9 00568        BLBC    R0, 58$                             ;
                                               18 A8  9F 0056B        PUSHAB  WINDOW_VALUE                        ; 0892
                                               28 AE  DD 0056E        PUSHL   DESC+4                              ; 0893
                                      7E       28 AE  3C 00571        MOVZWL  DESC, -(SP)                         ; 0892
```

```
                         00000000G  00           03  FB 00575         CALLS   #3, LIB$CVT_DTB                     ;
                                    14           50  E8 0057C         BLBS    R0, 56$                             ;
                                           20    AE  9F 0057F 53$:    PUSHAB  DESC                                ; 0897
                                                 01  DD 00582 54$:    PUSHL   #1                                  ;
                                     007710FA    8F  DD 00584         PUSHL   #7803130                            ;
                         00000000G  00           03  FB 0058A         CALLS   #3, LIB$SIGNAL                      ;
                                                 2E  11 00591 55$:    BRB     59$                                 ; 0898
                                    07     18    A8  D1 00593 56$:    CMPL    WINDOW_VALUE, #7                    ; 0900
                                                 0A  19 00597         BLSS    57$                                 ;
                         00000050   8F     18    A8  D1 00599         CMPL    WINDOW_VALUE, #80                   ; 0901
                                                 1A  15 005A1         BLEQ    58$                                 ;
                                     007711EA    8F  DD 005A3 57$:    PUSHL   #7803370                            ; 0904
                                           24    AE  9F 005A9         PUSHAB  DESC                                ;
                                                 01  DD 005AC         PUSHL   #1                                  ;
                                     007710FA    8F  DD 005AE         PUSHL   #7803130                            ;
                         00000000G  00           04  FB 005B4         CALLS   #4, LIB$SIGNAL                      ;
                                                 04  11 005BB         BRB     59$                                 ; 0905
                                    50           01  D0 005BD 58$:    MOVL    #1, R0                              ; 0910
                                                     04 005C0         RET                                         ;
                                                 50  D4 005C1 59$:    CLRL    R0                                  ; 0911
                                                     04 005C3         RET                                         ;

; Routine Size:  1476 bytes,    Routine Base:  $CODE$ + 0107
```

```
ROUTINE process_volume_set (root_desc, original_rvn, max_rvn) : NOVALUE =
BEGIN

!++
!
! Find each volume in the volume set and modify it.
!
! Inputs:
!       root_desc - descriptor of root volume
!       original_rvn - volume number of original volume
!       max_rvn = highest volume number in set
!
! Outputs:
!       None.
!
!--

MAP
    root_desc : REF VECTOR;


LOCAL
    status,
    status2,
    saved_flags,                                ! Saved original flags
    reduced_flags,                              ! Reduced flags
    this_rvn : volatile,
    iosb : VECTOR[4,WORD],                      ! $GETDVI status block
    desc1 : VECTOR[2],                          ! Device descriptors
    desc2 : VECTOR[2],
    buffer1 : VECTOR[128,BYTE],                 ! Device buffers
    buffer2 : VECTOR[128,BYTE],
    dvi_list : $ITMLST_DECL(ITEMS=2);           ! $GETDVI item list

!
! Do a little sneaky stuff first.  Transfer the root volume's name to the
! local descriptor.  Save the current flag settings, and calculate the
! flags for other volumes in this volume set.
!
desc1[0] = .root_desc[0];                       ! Set up so we
desc1[1] = buffer1;                             ! can loop easily
desc2[1] = buffer2;
CH$MOVE(.root_desc[0],
        .root_desc[1],
        buffer1);

saved_flags = .flags;                           ! Save original
reduced_flags = .flags AND                      ! The reduced set has
                (1^qual_erase OR                ! only the ERASE
                 1^qual_erase_val OR            ! and
                 1^qual_fhw OR                  ! HIGHWATER
                 1^qual_fhw_val);               ! qualifiers

!
! For each volume in the set, check to see if this is the original, or only
! one of the sister volumes, and set FLAGS accordingly.  To do this, we need
! to call $GETDVI to see what the volume number is.  But I'm getting ahead
```

```
  2 ! of myself...
  2 !
  2 WHILE true DO
  3     BEGIN
  3 !
  3 ! Open a file to the disk.
  3 !
  3     fab[fab$b_fns] = .desc1[0];
  3     fab[fab$l_fna] = .desc1[1];
  4     IF NOT (status = $OPEN(FAB = fab))
  3     THEN SIGNAL(set$_writeerr,                          ! Error accessing
  3                 1,
  3                 desc1,                                  ! this disk
  3                 .status)                                ! for this reason
  3     ELSE
  3         channel = .fab[fab$l_stv];
  3 !
  3 ! Get the next volume in the volume set, even if we can't use this one.
  3 ! If we can't even get to the next volume, then hang it up.
  3 !
P 3     $ITMLST_INIT(ITMLST = dvi_list,                     ! Set up DVI list
P 3                 (ITMCOD = dvi$_volnumber,               ! want current
P 3                  BUFADR = this_rvn),                    ! volume number,
P 3                 (ITMCOD = dvi$_nextdevnam,              ! next volume
P 3                  BUFADR = buffer2,                      ! name.
P 3                  BUFSIZ = %ALLOCATION(buffer2),
  3                  RETLEN = desc2));
P 3     status2 = $GETDVIW(ITMLST = dvi_list,               ! Get the info.
P 3                        DEVNAM = desc1,
  3                        IOSB   = iosb);
  3     IF .status2                                         ! If an error at
  3     THEN status2 = .iosb[0];                            ! this point, we
  3     IF NOT .status2                                     ! can't proceed,
  3     THEN                                                ! since we don't
  4         BEGIN                                           ! know which RVN
  4         SIGNAL(set$_writeerr,                           ! we're playing with
  4                1,
  4                desc1,
  4                .status2);
  4         RETURN;
  3         END;
  3
  3 !
  3 ! If the OPEN was successful, then process this volume.
  3 !
  3     IF .status
  3     THEN
  4         BEGIN
  4         IF .this_rvn NEQ .original_rvn              ! If not original disk,
  4         THEN flags = .reduced_flags;                ! use reduced flags.
  4         IF .flags NEQ 0                             ! If anything to set,
  4         THEN process_one_volume(desc1);             ! do it.
  4         flags = .saved_flags;                       ! Restore flags
  3         END;
  3
  3     $DASSGN(CHAN = .channel);                       ! Deassign the channel
```

```
; 1033          1026  3
; 1034          1027  3      ! Perform volume rebuild, if requested.
; 1035          1028  3      !
; 1036          1029  4      IF .status AND ( .this_rvn EQL .original_rvn )
; 1037          1030  3      THEN
; 1038          1031  3          IF .flags[qual_rebuild] AND .flags[qual_rebuild_val]
; 1039          1032  3          THEN
; 1040          1033  4              BEGIN
; 1041          1034  4
; 1042          1035  4              EXTERNAL ROUTINE
; 1043          1036  4                  stand_alone_rebuild;    ! Perform volume rebuild
; 1044          1037  4
; 1045          1038  4              LOCAL chan: WORD;
; 1046          1039  4
; 1047          1040  4              status = $ASSIGN( DEVNAM=desc1, CHAN=chan );
; 1048          1041  4              IF NOT .status
; 1049          1042  4              THEN SIGNAL (set$_openout, 1, desc1, .status, 0);
; 1050          1043  4
; 1051          1044  4              stand_alone_rebuild( .chan );        ! Do the rebuild.
; 1052          1045  4
; 1053          1046  4              status = $DASSGN( CHAN=.chan );
; 1054          1047  4              IF NOT .status
; 1055          1048  4              THEN SIGNAL (set$_closeout, 1, desc1, .status, 0);
; 1056          1049  4
; 1057          1050  3              END;
; 1058          1051  3
; 1059          1052  3      IF .this_rvn EQL .max_rvn                        ! If end of volume
; 1060          1053  3      THEN EXITLOOP;                                   ! set, leave
; 1061          1054  3
; 1062          1055  3      CH$MOVE(.desc2[0],                               ! Now switch to the
; 1063          1056  3              buffer2,                                 ! next volume in this
; 1064          1057  3              buffer1);                                ! volume set.
; 1065          1058  2      END;
; 1066          1059  2  !
; 1067          1060  2  !
; 1068          1061  2  ! For this volume set, if the /LABEL flag was set, then we must also
; 1069          1062  2  ! modify [0,0]VOLSET.SYS on the root volume.  Note that ODS1 volumes
; 1070          1063  2  ! cannot be volume sets so will fail this test.
; 1071          1064  2  !
; 1072          1065  2  IF .max_rvn GTR 1
; 1073          1066  2  AND .flags[qual_label]
; 1074          1067  2  THEN modify_volset(.root_desc);
; 1075          1068  2
; 1076          1069  2  RETURN;
; 1077          1070  1  END;



                                                            .EXTRN  SYS$OPEN, SYS$DASSGN
                                                            .EXTRN  STAND_ALONE_REBUILD
                                                            .EXTRN  SYS$ASSIGN

                                          0FFC 00000 PROCESS_VOLUME_SET:
                                                            .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11      ; 0912
                        5B 00000000'  EF  9E 00002          MOVAB   FLAGS, R11                                ;
                        5E      FEC4  CE  9E 00009          MOVAB   -316(SP), SP                              ;
                        56        04  AC  D0 0000E          MOVL    ROOT_DESC, R6                             ; 0951
```

```
                 EC    AD              66   D0 00012        MOVL     (R6), DESC1                            ;
                 F0    AD        FF64  CD   9E 00016        MOVAB    BUFFER1, DESC1+4                       ; 0952
                 E8    AD          20  AE   9E 0001C        MOVAB    BUFFER2, DESC2+4                       ; 0953
   FF64    CD    04    B6              66   28 00021        MOVC3    (R6), @4(R6), BUFFER1                  ; 0954
                       5A              6B   D0 00028        MOVL     FLAGS, SAVED_FLAGS                     ; 0958
           59    6B FFFF0FFF           8F   CB 0002B        BICL3    #-61441, FLAGS, REDUCED_FLAGS          ; 0960
              03BC    CB          EC   AD   90 00033 1$:    MOVB     DESC1, FAB+52                          ; 0977
              03B4    CB          F0   AD   D0 00039        MOVL     DESC1+4, FAB+44                        ; 0978
                                 0388  CB   9F 0003F        PUSHAB   FAB                                    ; 0979
          00000000G    00              01   FB 00043        CALLS    #1, SYS$OPEN                           ;
                       57              50   D0 0004A        MOVL     R0, STATUS                             ;
                       16              57   E8 0004D        BLBS     STATUS, 2$                             ;
                                       57   DD 00050        PUSHL    STATUS                                 ; 0983
                                  EC   AD   9F 00052        PUSHAB   DESC1                                  ; 0980
                                       01   DD 00055        PUSHL    #1                                     ;
                            00000000G  8F   DD 00057        PUSHL    #SET$_WRITEERR                         ;
          00000000G    00              04   FB 0005D        CALLS    #4, LIB$SIGNAL                         ;
                                       07   11 00064        BRB      3$                                     ;
              0224    CB         0394  CB   D0 00066 2$:    MOVL     FAB+12, CHANNEL                        ; 0985
                       50          04  AE   9E 0006D 3$:    MOVAB    DVI_LIST, $$ITMBLKPTR                  ; 0996
                       80  002E0004    8F   D0 00071        MOVL     #3014660, ($$ITMBLKPTR)+               ;
                       80          FC  AD   9E 00078        MOVAB    THIS_RVN, ($$ITMBLKPTR)+               ;
                                       80   D4 0007C        CLRL     ($$ITMBLKPTR)+                         ;
                       80  00340080    8F   D0 0007E        MOVL     #3408000, ($$ITMBLKPTR)+               ;
                       80          20  AE   9E 00085        MOVAB    BUFFER2, ($$ITMBLKPTR)+                ;
                       80          E4  AD   9E 00089        MOVAB    DESC2, ($$ITMBLKPTR)+                  ;
                                       80   D4 0008D        CLRL     ($$ITMBLKPTR)+                         ;
                                       7E   7C 0008F        CLRQ     -(SP)                                  ; 0999
                                       7E   D4 00091        CLRL     -(SP)                                  ;
                                  F4   AD   9F 00093        PUSHAB   IOSB                                   ;
                                  14   AE   9F 00096        PUSHAB   DVI_LIST                               ;
                                  EC   AD   9F 00099        PUSHAB   DESC1                                  ;
                                       7E   7C 0009C        CLRQ     -(SP)                                  ;
          00000000G    00              08   FB 0009E        CALLS    #8, SYS$GETDVIW                        ;
                       58              50   D0 000A5        MOVL     R0, STATUS2                            ;
                       07              58   E9 000A8        BLBC     STATUS2, 4$                            ; 1000
                       58          F4  AD   3C 000AB        MOVZWL   IOSB, STATUS2                          ; 1001
                       15              58   E8 000AF        BLBS     STATUS2, 5$                            ; 1002
                                       58   DD 000B2 4$:    PUSHL    STATUS2                                ; 1008
                                  EC   AD   9F 000B4        PUSHAB   DESC1                                  ; 1005
                                       01   DD 000B7        PUSHL    #1                                     ;
                            00000000G  8F   DD 000B9        PUSHL    #SET$_WRITEERR                         ;
          00000000G    00              04   FB 000BF        CALLS    #4, LIB$SIGNAL                         ;
                                            04 000C6        RET                                             ; 1004
                       1B              57   E9 000C7 5$:    BLBC     STATUS, 8$                             ; 1015
                 08    AC          FC  AD   D1 000CA        CMPL     THIS_RVN, ORIGINAL_RVN                 ; 1018
                                       03   13 000CF        BEQL     6$                                     ;
                       6B              59   D0 000D1        MOVL     REDUCED_FLAGS, FLAGS                   ; 1019
                                       6B   D5 000D4 6$:    TSTL     FLAGS                                  ; 1020
                                       0A   13 000D6        BEQL     7$                                     ;
                                  EC   AD   9F 000D8        PUSHAB   DESC1                                  ; 1021
          00000000V    EF              01   FB 000DB        CALLS    #1, PROCESS_ONE_VOLUME                 ;
                       6B              5A   D0 000E2 7$:    MOVL     SAVED_FLAGS, FLAGS                     ; 1022
                                 0224  CB   DD 000E5 8$:    PUSHL    CHANNEL                                ; 1025
          00000000G    00              01   FB 000E9        CALLS    #1, SYS$DASSGN                         ;
                       6C              57   E9 000F0        BLBC     STATUS, 10$                            ; 1029
                 08    AC          FC  AD   D1 000F3        CMPL     THIS_RVN, ORIGINAL_RVN                 ;
```

```
                                            65  12 000F8          BNEQ    10$                                   ;
         60       02  AB                    04  E1 000FA          BBC     #4, FLAGS+2, 10$                      ; 1031
         5B       02  AB                    05  E1 000FF          BBC     #5, FLAGS+2, 10$                      ;
                                            7E  7C 00104          CLRQ    -(SP)                                 ; 1040
                                   08  AE  9F 00106          PUSHAB  CHAN                                  ;
                                   EC  AD  9F 00109          PUSHAB  DESC1                                 ;
              00000000G  00                 04  FB 0010C          CALLS   #4, SYS$ASSIGN                        ;
                         57                 50  D0 00113          MOVL    R0, STATUS                            ;
                         16                 57  E8 00116          BLBS    STATUS, 9$                            ; 1041
                                            7E  D4 00119          CLRL    -(SP)                                 ; 1042
                                            57  DD 0011B          PUSHL   STATUS                                ;
                                   EC  AD  9F 0011D          PUSHAB  DESC1                                 ;
                                            01  DD 00120          PUSHL   #1                                    ;
                             007710A2  8F  DD 00122          PUSHL   #7803042                              ;
              00000000G  00                 05  FB 00128          CALLS   #5, LIB$SIGNAL                        ;
                         7E                 6E  3C 0012F  9$:     MOVZWL  CHAN, -(SP)                           ; 1044
              00000000G  00                 01  FB 00132          CALLS   #1, STAND_ALONE_REBUILD               ;
                         7E                 6E  3C 00139          MOVZWL  CHAN, -(SP)                           ; 1046
              00000000G  00                 01  FB 0013C          CALLS   #1, SYS$DASSGN                        ;
                         57                 50  D0 00143          MOVL    R0, STATUS                            ;
                         16                 57  E8 00146          BLBS    STATUS, 10$                           ; 1047
                                            7E  D4 00149          CLRL    -(SP)                                 ; 1048
                                            57  DD 0014B          PUSHL   STATUS                                ;
                                   EC  AD  9F 0014D          PUSHAB  DESC1                                 ;
                                            01  DD 00150          PUSHL   #1                                    ;
                             0077105A  8F  DD 00152          PUSHL   #7802970                              ;
              00000000G  00                 05  FB 00158          CALLS   #5, LIB$SIGNAL                        ;
                     0C  AC            FC  AD  D1 0015F  10$:    CMPL    THIS_RVN, MAX_RVN                     ; 1052
                                            0B  13 00164          BEQL    11$                                   ;
   FF64   CD        20  AE            E4  AD  28 00166          MOVC3   DESC2, BUFFER2, BUFFER1               ; 1055
                                          FEC2  31 0016E          BRW     1$                                    ; 0971
                         01            0C  AC  D1 00171  11$:    CMPL    MAX_RVN, #1                           ; 1065
                                            0D  15 00175          BLEQ    12$                                   ;
         09               6B                05  E1 00177          BBC     #5, FLAGS, 12$                        ; 1066
                                            56  DD 0017B          PUSHL   R6                                    ; 1067
              00000000V  EF                 01  FB 0017D          CALLS   #1, MODIFY_VOLSET                     ;
                                                04 00184  12$:    RET                                           ; 1070

; Routine Size:  389 bytes,    Routine Base:  $CODE$ + 06CB
```

```
1 ROUTINE process_one_volume (desc) : NOVALUE =
2 BEGIN
2
2 !++
2 !
2 ! Find each volume in the volume set and call the routines which
2 ! actually modify the data.
2 !
2 ! Inputs:
2 !       desc - address of volume descriptor
2 !
2 ! Outputs:
2 !       None.  The volumes and I/O database are modified.
2 !
2 !---
2
2 LOCAL
2     status,
2     vbn,                                      ! Place to store vbn
2     ucb,                                      ! Place to store UCB address
2     cluster;                                  ! Cluster size of volume
2
2 IF NOT read_homeblock(cluster)                ! If can't read home block
2 THEN SIGNAL(set$_nohome)                      ! then tell the user
2 ELSE
3     BEGIN
3     status = 0;                               ! Show that no homeblocks modified yet
3     INCR vbn FROM 2 TO .cluster*3 DO          ! Go thru all homeblocks on the volume
4         BEGIN
4 !
4 ! Call the routine that reads, modifies, and writes the homeblock.  If
4 ! successful, set STATUS = 1
4 !
4         IF set_home(.vbn, .desc)
4         THEN status = 1;
4         IF .ods1
4         THEN EXITLOOP;                        ! Finished for ODS1
3         END;
3 !
3 ! If STATUS = 1, then at least some of the homeblocks were good and were
3 ! modified.
3 !
3     IF .status THEN
4         BEGIN
4         ! So, go ahead and change the I/O database.
4         !
4         ucb = KERNEL_CALL (GET_CHANNELUCB, .channel);
4         KERNEL_CALL(SET_UCBVCB, .ucb);
4
4         IF .flags[qual_log]                   ! If /LOG, tell user
4         THEN SIGNAL (set$_modified, 1, .desc);
3         END;
2     END;
2
2 RETURN;
1 END;
```

```
                                                                   .EXTRN  SYS$CMKRNL

                                            00FC 00000 PROCESS_ONE_VOLUME:
                                                                   .WORD   Save R2,R3,R4,R5,R6,R7                          ; 1071
                         57 00000000G  9F    9E 00002              MOVAB   @#SYS$CMKRNL, R7                                ;
                         56 00000000G  00    9E 00009              MOVAB   LIB$SIGNAL, R6                                  ;
                         55 00000000'  EF    9E 00010              MOVAB   ODS1, R5                                        ;
                         5E            04    C2 00017              SUBL2   #4, SP                                          ;
                                       5E    DD 0001A              PUSHL   SP                                              ; 1093
               00000000V EF            01    FB 0001C              CALLS   #1, READ_HOMEBLOCK                              ;
                         0A            50    E8 00023              BLBS    R0, 1$                                          ;
                            00000000G  8F    DD 00026              PUSHL   #SET$_NOHOME                                    ; 1094
                         66            01    FB 0002C              CALLS   #1, LIB$SIGNAL                                  ;
                                             04 0002F              RET                                                     ;
                                       54    D4 00030 1$:          CLRL    STATUS                                          ; 1097
         53              6E            03    C5 00032              MULL3   #3, CLUSTER, R3                                 ; 1098
                         52            01    D0 00036              MOVL    #1, VBN                                         ; 1104
                                       15    11 00039              BRB     4$                                              ;
                                   04  AC    DD 0003B 2$:          PUSHL   DESC                                            ;
                                       52    DD 0003E              PUSHL   VBN                                             ;
               00000000V EF            02    FB 00040              CALLS   #2, SET_HOME                                    ;
                         03            50    E9 00047              BLBC    R0, 3$                                          ;
                         54            01    D0 0004A              MOVL    #1, STATUS                                      ; 1105
                         04            65    E8 0004D 3$:          BLBS    ODS1, 5$                                        ; 1106
         E7              52            53    F3 00050 4$:          AOBLEQ  R3, VBN, 2$                                     ; 1098
                         33            54    E9 00054 5$:          BLBC    STATUS, 6$                                      ; 1113
                                   03  A5    DD 00057              PUSHL   CHANNEL                                         ; 1117
                                       01    DD 0005A              PUSHL   #1                                              ;
                                       5E    DD 0005C              PUSHL   SP                                              ;
                            00000000G  00    9F 0005E              PUSHAB  GET_CHANNELUCB                                  ;
                         67            04    FB 00064              CALLS   #4, SYS$CMKRNL                                  ;
                                       50    DD 00067              PUSHL   UCB                                             ; 1118
                                       01    DD 00069              PUSHL   #1                                              ;
                                       5E    DD 0006B              PUSHL   SP                                              ;
                            00000000V  EF    9F 0006D              PUSHAB  SET_UCBVCB                                      ;
                         67            04    FB 00073              CALLS   #4, SYS$CMKRNL                                  ;
         0E        FDDF  C5            06    E1 00076              BBC     #6, FLAGS, 6$                                   ; 1120
                                   04  AC    DD 0007C              PUSHL   DESC                                            ; 1121
                                       01    DD 0007F              PUSHL   #1                                              ;
                            00000000G  8F    DD 00081              PUSHL   #SET$_MODIFIED                                  ;
                         66            03    FB 00087              CALLS   #3, LIB$SIGNAL                                  ;
                                             04 0008A 6$:          RET                                                     ; 1126

; Routine Size:  139 bytes,    Routine Base:  $CODE$ + 0850
```

```
  1 ROUTINE read_homeblock(cluster) =
  1 !++
  1 !
  1 ! This routine reads the first good home block of the volume.
  1 ! It uses $QIOW's because $READ finds the End-of-File block to be
  1 ! zero in ODS1 initialized disks and thus will not try to read the home block.
  1 ! In addition the cluster size and structure level are determined and stored.
  1 !
  1 ! Outputs:
  1 !       cluster - cluster size
  1 !       ods1 - 0 => ODS2
  1 !              1 => ODS1
  1 !
  1 !--
  2 BEGIN
  2
  2 LOCAL
  2     desc : $BBLOCK[dsc$c_s_bln],         ! Descriptor for the FIB in $QIOW
  2     fib  : $BBLOCK[fib$c_extdata],       ! File Information Block for $QIOW
  2     atr  : BLOCKVECTOR[2,8,BYTE],        ! Attribute list for $QIOW
  2     stablk : $BBLOCK[32],                ! Where statistics block is stored after $QIOW
  2     file_size : VECTOR[2,WORD],          ! The file size from statistics block
  2     iosb : VECTOR[4,WORD],               ! Status block for the $QIOW
  2     block,                               ! Temporary block count
  2     status;                              ! Status
  2
  2 ! Before we can look at the homeblock we have to find how many blocks there
  2 ! are (or the block number or the last block).  This is done by issuing a
  2 ! $QIOW to get the statistics block.
  2
  2 desc[dsc$w_length] = fib$c_extdata;      ! Initialize descriptor pointing to
  2 desc[dsc$a_pointer] = fib;                ! to the file info block
  2
  2 CH$FILL(0, fib$c_extdata, fib);          ! Zero the fib for new info
  2
  2 fib[fib$l_acctl] = fib$m_noread OR       ! Deny read and write access to others
  2                    fib$m_nowrite;
  2 fib[fib$w_fid_num] = .nam[nam$w_fid_num];
  2 fib[fib$w_fid_seq] = .nam[nam$w_fid_seq];  ! Specify file identification
  2 fib[fib$w_fid_rvn] = .nam[nam$w_fid_rvn];
  2
  2 atr[0,atr$w_type] = atr$c_statblk;       ! The attribute we want is the
  2 atr[0,atr$w_size] = atr$s_statblk;       ! statistics block
  2 atr[0,atr$l_addr] = stablk;              ! It goes into stablk
  2 atr[1,0,0,32,0]   = 0;                   ! Indicate end of information
  2
P 2 status = $QIOW (CHAN = .channel,         ! Access the statistics block
P 2                 FUNC = IO$_ACCESS,
P 2                 IOSB = iosb,
P 2                 P1   = desc,
  2                 P5   = atr);
  2 IF  .status THEN status = .iosb[0];      ! Check if everything Okay
  2 IF NOT .status
  2 THEN SIGNAL(.status)                     ! If not, tell user, go to end
  2 ELSE                                     ! If okay
  3 BEGIN
  3     file_size[1] = .stablk[sbk$w_filesizh];  ! The file size is stored
```

```
    file_size[0] = .stablk[sbk$w_filesizl];  ! backwards so invert
!
! It is possible the homeblock exists so . . .
! Keep reading until we get a block that reads without errors and meets the
! criteria for a homeblock.
!
    INCR block FROM 2 TO 100 DO
        BEGIN
        IF .block LEQ .file_size                  ! If we have not passed the end of file
        THEN
        BEGIN
            status = $QIOW (CHAN = .channel,      ! Read the virtual 'block'
                            FUNC = IO$_READVBLK,
                            IOSB = iosb,
                            P1   = buffer,        ! Put it in 'buffer'
                            P2   = 512,           ! Get a whole block
                            P3   = .block);
            IF  .status THEN status = .iosb[0];
            IF NOT .status
            THEN
            BEGIN
                SIGNAL(.status);
                RETURN false;
            END;
            IF
                .buffer[hm2$b_struclev] EQL 2 AND
                .buffer[hm2$l_altidxlbn] NEQ 0 AND
                .buffer[hm2$w_cluster] NEQ 0 AND
                .buffer[hm2$w_homevbn] NEQ 0 AND
                .buffer[hm2$w_alhomevbn] NEQ 0 AND
                .buffer[hm2$w_altidxvbn] NEQ 0 AND
                .buffer[hm2$w_ibmapvbn] NEQ 0 AND
                .buffer[hm2$l_ibmaplbn] NEQ 0 AND
                .buffer[hm2$l_maxfiles] NEQ 0 AND
                .buffer[hm2$w_ibmapsize] NEQ 0 AND
                .buffer[hm2$w_resfiles] NEQ 0 AND
                checksum2(buffer, $BYTEOFFSET(hm2$w_checksum1)) AND
                checksum2(buffer, $BYTEOFFSET(hm2$w_checksum2))
            THEN
            BEGIN
                ods1 = 0;                               ! This is an ODS2 volume
                .cluster = .buffer[hm2$w_cluster];      ! with this cluster size
                IF .flags[qual_access]  ! If /ACCESSED was specified,
                THEN                                    ! compute the value to add
                BEGIN                                   ! to the LRU value in the VCB
                    acc_inc = 0;
                    IF .acc_value GTR .buffer[hm2$b_lru_lim]
                    THEN acc_inc = .acc_value - .buffer[hm2$b_lru_lim];
                END;
                RETURN true;
            END
                ELSE IF
                    .buffer[hm1$w_struclev] EQL hm1$c_level1 AND
                    .buffer[hm1$w_cluster] NEQ 0 AND
                    .buffer[hm1$l_ibmaplbn] NEQ 0 AND
                    .buffer[hm1$w_maxfiles] NEQ 0 AND
```

```
; 1250          1241  5                   .buffer[hm1$w_ibmapsize] NEQ 0 AND
; 1251          1242  5                   checksum2(buffer, $BYTEOFFSET(hm1$w_checksum1)) AND
; 1252          1243  5                   checksum2(buffer, $BYTEOFFSET(hm1$w_checksum2))
; 1253          1244  5               THEN
; 1254          1245  6                   BEGIN
; 1255          1246  6                   ods1 = 1;                               ! Volume is ODS1
; 1256          1247  6                   .cluster = 1;                           ! Dummy in a cluster size of 1
; 1257          1248  6                   IF .flags[qual_access]      ! If /ACCESSED was specified,
; 1258          1249  6                   THEN                                    ! compute the value to add
; 1259          1250  7                       BEGIN                               ! to the LRU value in the VCB
; 1260          1251  7                       acc_inc = 0;
; 1261          1252  7                       IF .acc_value GTR .buffer[hm1$b_lru_lim]
; 1262          1253  7                       THEN acc_inc = .acc_value - .buffer[hm1$b_lru_lim];
; 1263          1254  6                       END;
; 1264          1255  6                   RETURN true;
; 1265          1256  5                   END;
; 1266          1257  4               END;                                        ! End of read success block
; 1267          1258  3           END;                                            ! End of INC block
; 1268          1259  2       END;                                                ! End of file access block
; 1269          1260  2 !
; 1270          1261  2 ! If here, then no good homeblock was found. Return a value of FALSE to
; 1271          1262  2 ! show that.
; 1272          1263  2 !
; 1273          1264  2 RETURN false;
; 1274          1265  1 END;



                                                                        .EXTRN  SYS$QIOW

                                                03FC 00000 READ_HOMEBLOCK:
                                                                        .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9          ; 1127
                            59 00000000G  00  9E 00002          MOVAB   SYS$QIOW, R9                          ;
                            58 00000000'  EF  9E 00009          MOVAB   ACC_VALUE, R8                         ;
                            57 00000000G  00  9E 00010          MOVAB   CHECKSUM2, R7                         ;
                            56 00000000'  EF  9E 00017          MOVAB   BUFFER, R6                            ;
                            5E        9C  AE  9E 0001E          MOVAB   -100(SP), SP                          ;
                    5C  AE            20  B0 00022          MOVW    #32, DESC                             ; 1157
                    60  AE        3C  AE  9E 00026          MOVAB   FIB, DESC+4                           ; 1158
     20            00       6E        00  2C 0002B          MOVC5   #0, (SP), #0, #32, FIB                ; 1160
                                  3C  AE     00030                                                        ;
                    3C  AE      0401  8F  3C 00032          MOVZWL  #1025, FIB                            ; 1162
                    40  AE      032C  C6  D0 00038          MOVL    NAM+36, FIB+4                         ; 1164
                    44  AE      0330  C6  B0 0003E          MOVW    NAM+40, FIB+8                         ; 1166
                    2C  AE  00090020  8F  D0 00044          MOVL    #589856, ATR                          ; 1169
                    30  AE        0C  AE  9E 0004C          MOVAB   STABLK, ATR+4                         ; 1170
                                  34  AE  D4 00051          CLRL    ATR+8                                 ; 1171
                                      7E  D4 00054          CLRL    -(SP)                                 ; 1177
                                  30  AE  9F 00056          PUSHAB  ATR                                   ;
                                      7E  7C 00059          CLRQ    -(SP)                                 ;
                                      7E  D4 0005B          CLRL    -(SP)                                 ;
                                  70  AE  9F 0005D          PUSHAB  DESC                                  ;
                                      7E  7C 00060          CLRQ    -(SP)                                 ;
                                  24  AE  9F 00062          PUSHAB  IOSB                                  ;
                                      32  DD 00065          PUSHL   #50                                   ;
                                0204  C6  DD 00067          PUSHL   CHANNEL                               ;
                                      7E  D4 0006B          CLRL    -(SP)                                 ;
```

```
                       69              0C  FB 0006D        CALLS   #12, SYS$QIOW                    ;
                       53              50  D0 00070        MOVL    R0, STATUS                       ;
                       07              53  E9 00073        BLBC    STATUS, 1$                       ; 1178
                       53      04      AE  3C 00076        MOVZWL  IOSB, STATUS                     ;
                       0C              53  E8 0007A        BLBS    STATUS, 2$                       ; 1179
                                       53  DD 0007D 1$:    PUSHL   STATUS                           ; 1180
            00000000G  00              01  FB 0007F        CALLS   #1, LIB$SIGNAL                   ;
                                     0111  31 00086        BRW     8$                               ;
                   02  AE      10      AE  B0 00089 2$:    MOVW    STABLK+4, FILE_SIZE+2            ; 1183
                       6E      12      AE  B0 0008E        MOVW    STABLK+6, FILE_SIZE              ; 1184
                       52              02  D0 00092        MOVL    #2, BLOCK                        ; 1193
                       6E              52  D1 00095 3$:    CMPL    BLOCK, FILE_SIZE                 ;
                                       03  15 00098        BLEQ    4$                               ;
                                     00F3  31 0009A        BRW     7$                               ;
                                       7E  7C 0009D 4$:    CLRQ    -(SP)                            ; 1201
                                       7E  D4 0009F        CLRL    -(SP)                            ;
                                       52  DD 000A1        PUSHL   BLOCK                            ;
                       7E    0200      8F  3C 000A3        MOVZWL  #512, -(SP)                      ;
                                       56  DD 000A8        PUSHL   R6                               ;
                                       7E  7C 000AA        CLRQ    -(SP)                            ;
                               24      AE  9F 000AC        PUSHAB  IOSB                             ;
                                       31  DD 000AF        PUSHL   #49                              ;
                             0204      C6  DD 000B1        PUSHL   CHANNEL                          ;
                                       7E  D4 000B5        CLRL    -(SP)                            ;
                       69              0C  FB 000B7        CALLS   #12, SYS$QIOW                    ;
                       53              50  D0 000BA        MOVL    R0, STATUS                       ;
                       BD              53  E9 000BD        BLBC    STATUS, 1$                       ; 1202
                       53      04      AE  3C 000C0        MOVZWL  IOSB, STATUS                     ;
                       B6              53  E9 000C4        BLBC    STATUS, 1$                       ; 1203
                       02      0D      A6  91 000C7        CMPB    BUFFER+13, #2                    ; 1210
                                       6C  12 000CB        BNEQ    5$                               ;
                               08      A6  D5 000CD        TSTL    BUFFER+8                         ; 1211
                                       67  13 000D0        BEQL    5$                               ;
                               0E      A6  B5 000D2        TSTW    BUFFER+14                        ; 1212
                                       62  13 000D5        BEQL    5$                               ;
                               10      A6  B5 000D7        TSTW    BUFFER+16                        ; 1213
                                       5D  13 000DA        BEQL    5$                               ;
                               12      A6  B5 000DC        TSTW    BUFFER+18                        ; 1214
                                       58  13 000DF        BEQL    5$                               ;
                               14      A6  B5 000E1        TSTW    BUFFER+20                        ; 1215
                                       53  13 000E4        BEQL    5$                               ;
                               16      A6  B5 000E6        TSTW    BUFFER+22                        ; 1216
                                       4E  13 000E9        BEQL    5$                               ;
                               18      A6  D5 000EB        TSTL    BUFFER+24                        ; 1217
                                       49  13 000EE        BEQL    5$                               ;
                               1C      A6  D5 000F0        TSTL    BUFFER+28                        ; 1218
                                       44  13 000F3        BEQL    5$                               ;
                               20      A6  B5 000F5        TSTW    BUFFER+32                        ; 1219
                                       3F  13 000F8        BEQL    5$                               ;
                               22      A6  B5 000FA        TSTW    BUFFER+34                        ; 1220
                                       3A  13 000FD        BEQL    5$                               ;
                                       3A  DD 000FF        PUSHL   #58                              ; 1221
                                       56  DD 00101        PUSHL   R6                               ;
                       67              02  FB 00103        CALLS   #2, CHECKSUM2                    ;
                       30              50  E9 00106        BLBC    R0, 5$                           ;
                       7E    01FE      8F  3C 00109        MOVZWL  #510, -(SP)                      ; 1222
                                       56  DD 0010E        PUSHL   R6                               ;
```

```
                            67            02  FB 00110           CALLS   #2, CHECKSUM2                          ;
                            23            50  E9 00113           BLBC    R0, 5$                                 ;
                                   0201   C6  94 00116           CLRB    ODS1                                   ; 1225
                       04   BC       0E   A6  3C 0011A           MOVZWL  BUFFER+14, @CLUSTER                    ; 1226
                 68    E0   A6            01  E1 0011F           BBC     #1, FLAGS, 6$                          ; 1227
                                   0200   C6  94 00124           CLRB    ACC_INC                                ; 1230
    68     45    A6         08            00  ED 00128           CMPZV   #0, #8, BUFFER+69, ACC_VALUE           ; 1231
                                          5C  18 0012E           BGEQ    6$                                     ;
         0200    C6         68       45   A6  83 00130           SUBB3   BUFFER+69, ACC_VALUE, ACC_INC          ; 1232
                                          53  11 00137           BRB     6$                                     ; 1234
                 0101  8F            0C   A6  B1 00139 5$:       CMPW    BUFFER+12, #257                        ; 1237
                                          4F  12 0013F           BNEQ    7$                                     ;
                                     08   A6  B5 00141           TSTW    BUFFER+8                               ; 1238
                                          4A  13 00144           BEQL    7$                                     ;
                                     02   A6  D5 00146           TSTL    BUFFER+2                               ; 1239
                                          45  13 00149           BEQL    7$                                     ;
                                     06   A6  B5 0014B           TSTW    BUFFER+6                               ; 1240
                                          40  13 0014E           BEQL    7$                                     ;
                                          66  B5 00150           TSTW    BUFFER                                 ; 1241
                                          3C  13 00152           BEQL    7$                                     ;
                                          3A  DD 00154           PUSHL   #58                                    ; 1242
                                          56  DD 00156           PUSHL   R6                                     ;
                            67            02  FB 00158           CALLS   #2, CHECKSUM2                          ;
                            32            50  E9 0015B           BLBC    R0, 7$                                 ;
                            7E     01FE   8F  3C 0015E           MOVZWL  #510, -(SP)                            ; 1243
                                          56  DD 00163           PUSHL   R6                                     ;
                            67            02  FB 00165           CALLS   #2, CHECKSUM2                          ;
                            25            50  E9 00168           BLBC    R0, 7$                                 ;
                 0201  C6                 01  90 0016B           MOVB    #1, ODS1                               ; 1246
                   04  BC                 01  D0 00170           MOVL    #1, @CLUSTER                           ; 1247
                 13    E0   A6            01  E1 00174           BBC     #1, FLAGS, 6$                          ; 1248
                                   0200   C6  94 00179           CLRB    ACC_INC                                ; 1251
    68     2E    A6         08            00  ED 0017D           CMPZV   #0, #8, BUFFER+46, ACC_VALUE           ; 1252
                                          07  18 00183           BGEQ    6$                                     ;
         0200    C6         68       2E   A6  83 00185           SUBB3   BUFFER+46, ACC_VALUE, ACC_INC          ; 1253
                            50            01  D0 0018C 6$:       MOVL    #1, R0                                 ; 1255
                                              04 0018F           RET                                            ;
  FEFB           52         01 00000064   8F  F1 00190 7$:       ACBL    #100, #1, BLOCK, 3$                    ; 1191
                                          50  D4 0019A 8$:       CLRL    R0                                     ; 1265
                                              04 0019C           RET                                            ;

; Routine Size:  413 bytes,    Routine Base:  $CODE$ + 08DB
; 1276          1266  1 ROUTINE parse_class (desc) =
; 1277          1267  2 BEGIN
; 1278          1268  2
; 1279          1269  2 !---
; 1280          1270  2 !
; 1281          1271  2 ! This routine parses one class of user (e.g. SYSTEM, OWNER, GROUP, WORLD)
; 1282          1272  2 ! to see what protection is allowed.  The value returned in the low 4 bits
; 1283          1273  2 ! is the protection code, with the bits set to reflect that access is
; 1284          1274  2 ! requested.  Note that this is exactly the opposite of what the system wants.
; 1285          1275  2 !
; 1286          1276  2 ! Inputs:
; 1287          1277  2 !
; 1288          1278  2 !       DESC - a descriptor pointing to the ASCII representation of the
; 1289          1279  2 !              protection desired
; 1290          1280  2 !
; 1291          1281  2 !---
; 1292          1282  2
; 1293          1283  2 MAP desc : REF $BBLOCK;
; 1294          1284  2
; 1295          1285  2 LOCAL
; 1296          1286  2     result,
; 1297          1287  2     string : REF VECTOR[,BYTE];          ! String pointer
; 1298          1288  2
; 1299          1289  2 !
; 1300          1290  2 ! Initially set the value to all zeros, no access
; 1301          1291  2 !
; 1302          1292  2 result = 0;
; 1303          1293  2
; 1304          1294  2 !
; 1305          1295  2 ! Scan for the occurrence of each keyletter, and, if it is there, set the
; 1306          1296  2 ! appropriate bit.
; 1307          1297  2 !
; 1308          1298  2 string = .desc[dsc$a_pointer];
; 1309          1299  2 INCR index FROM 0 to (.desc[dsc$w_length] -1) DO
; 1310          1300  3     BEGIN
; 1311          1301  3     IF .string[.index] EQL'R'
; 1312          1302  3     THEN result = .result OR %X'1'
; 1313          1303  3     ELSE IF .string[.index] EQL  'W'
; 1314          1304  3     THEN result = .result OR %X'2'
; 1315          1305  3     ELSE IF .string[.index] EQL 'E'
; 1316          1306  3          OR .string[.index] EQL 'P'
; 1317          1307  3     THEN result = .result OR %X'4'
; 1318          1308  3     ELSE IF .string[.index] EQL 'D'
; 1319          1309  3          OR .string[.index] EQL 'L'
; 1320          1310  3     THEN result = .result OR %X'8'
; 1321          1311  3     ELSE SIGNAL_STOP(cli$_ivprot);
; 1322          1312  2     END;
; 1323          1313  2
; 1324          1314  2 RETURN .result;
; 1325          1315  1 END;



                                                003C 00000 PARSE_CLASS:
                                                                .WORD   Save R2,R3,R4,R5                                    ; 1266
```

```
                                    54  D4 00002         CLRL    RESULT                                    ; 1292
              50         04         AC  D0 00004         MOVL    DESC, R0                                  ; 1298
              52         04         A0  D0 00008         MOVL    4(R0), STRING                             ;
              55                    60  3C 0000C         MOVZWL  (R0), R5                                  ; 1299
              53                    01  CE 0000F         MNEGL   #1, INDEX                                 ; 1301
                                    49  11 00012         BRB     8$                                        ;
              50                  6342  9A 00014 1$:     MOVZBL  (INDEX)[STRING], R0                       ;
         52   8F                    50  91 00018         CMPB    R0, #82                                   ;
                                    05  12 0001C         BNEQ    2$                                        ;
              54                    01  88 0001E         BISB2   #1, RESULT                                ; 1302
                                    3A  11 00021         BRB     8$                                        ;
         57   8F                    50  91 00023 2$:     CMPB    R0, #87                                   ; 1303
                                    05  12 00027         BNEQ    3$                                        ;
              54                    02  88 00029         BISB2   #2, RESULT                                ; 1304
                                    2F  11 0002C         BRB     8$                                        ;
         45   8F                    50  91 0002E 3$:     CMPB    R0, #69                                   ; 1305
                                    06  13 00032         BEQL    4$                                        ;
         50   8F                    50  91 00034         CMPB    R0, #80                                   ; 1306
                                    05  12 00038         BNEQ    5$                                        ;
              54                    04  88 0003A 4$:     BISB2   #4, RESULT                                ; 1307
                                    1E  11 0003D         BRB     8$                                        ;
         44   8F                    50  91 0003F 5$:     CMPB    R0, #68                                   ; 1308
                                    06  13 00043         BEQL    6$                                        ;
         4C   8F                    50  91 00045         CMPB    R0, #76                                   ; 1309
                                    05  12 00049         BNEQ    7$                                        ;
              54                    08  88 0004B 6$:     BISB2   #8, RESULT                                ; 1310
                                    0D  11 0004E         BRB     8$                                        ;
                         00000000G  8F  DD 00050 7$:     PUSHL   #CLI$_IVPROT                              ; 1311
      00000000G  00                 01  FB 00056         CALLS   #1, LIB$STOP                              ;
   B3            53                 55  F2 0005D 8$:     AOBLSS  R5, INDEX, 1$                             ; 1299
                 50                 54  D0 00061         MOVL    RESULT, R0                                ; 1314
                                        04 00064         RET                                               ; 1315

; Routine Size:  101 bytes,    Routine Base:  $CODE$ + 0A78
```

```
  1 ROUTINE set_home (vbn, desc) =
  1 !++
  1 !
  1 ! This routine reads a homeblock, modifies it, and writes it back to the
  1 ! volume.
  1 !
  1 ! Inputs:
  1 !       vbn - current vbn to read
  1 !       ods1 - 0 => ODS2
  1 !              1 => ODS1
  1 !       desc - descriptor for the volume
  1 !
  1 !--
  2 BEGIN
  2
  2
  2 LOCAL
  2     iosb : VECTOR[4,WORD],          ! I/O Status Block for $QIOW
  2     status;                         ! General status return
  2
  2 !
  2 ! Read the homeblock.
  2 !
P 2 status = $QIOW (CHAN = .channel,
P 2                 FUNC = IO$_READVBLK,    ! Read virtual block
P 2                 IOSB = iosb,
P 2                 P1   = buffer,          ! Place it in 'buffer'
P 2                 P2   = 512,             ! Read 512 bytes
  2                 P3   = .vbn);           ! Starting at this virutal block
  2 IF .status THEN status = .iosb[0];
  2 IF NOT .status
  2 THEN
  3     BEGIN
  3     SIGNAL(set$_hbread,             ! Error reading homeblock
  3             1,
  3             .desc,                  ! for this volume
  3             .status);               ! for this reason
  3     RETURN false;
  2     END;
  2
  2 !
  2 ! Change the ACCESSED (LRU) value, if requested
  2 !
  2 IF .flags[qual_access]
  2 THEN
  2     IF .ods1 THEN buffer[hm1$b_lru_lim] = .acc_value    ! For ODS1
  2     ELSE buffer[hm2$b_lru_lim] = .acc_value;            ! For ODS2
  2
  2 !
  2 ! If the DATA_CHECK qualifier is set, check to see if ODS1 or ODS2.  If ODS1,
  2 ! tell the user that DATA_CHECK is illegal.  Otherwise, set the bits.
  2 !
  2 IF .flags[qual_data]
  2 THEN IF .ods1                                          ! If ODS1,
  2 THEN SIGNAL(set$_notods2,                             ! tell user no
  2             1,
  2             $DESCRIPTOR('DATA_CHECK'))
```

```
2 ELSE
3     BEGIN
3     IF .dflags[data_read] THEN buffer[hm2$v_readcheck] = 1;
3     IF .dflags[data_noread] THEN buffer[hm2$v_readcheck] = 0;
3     IF .dflags[data_write] THEN buffer[hm2$v_writcheck] = 1;
3     IF .dflags[data_nowrite] THEN buffer[hm2$v_writcheck] = 0;
2     END;
2
2 !
2 ! [NO]ERASE_ON_DELETE only works for ODS2.
2 !
2 IF .flags[qual_erase]
2 THEN IF .ods1
2 THEN SIGNAL(set$_notods2, 1, %ASCID 'ERASE_ON_DELETE')
2 ELSE buffer[hm2$v_erase] = .flags[qual_erase_val];
2
2 !
2 ! For the EXTENSION qualifier, if ODS1, the field is only a byte long, so
2 ! the greatest value is 255.  If the user specified a larger value, tell the
2 ! user and return.  Otherwise, make the change.
2 !
2 IF .flags[qual_exte]
2 THEN IF .ods1
2 THEN
3     BEGIN                                                   ! Start of ODS1
3     IF .exte_value GTR 255
3     THEN
4         BEGIN
4         SIGNAL(set$_valerr);
4         RETURN false;
4         END
3     ELSE buffer[hm1$b_extend] = .exte_value;
3     END                                                     ! End of ODS1
2 ELSE buffer[hm2$w_extend] = .exte_value;                   ! Change ODS2 extend
2
2 !
2 ! For FILE_PROTECTION, the location is different, depending on which type of
2 ! disk we have.
2 !
2 ! Also, a word about the value in FPROT_VALUE.  The high word,
2 ! FPROT_VALUE<16,16>, contains a mask indicating which groups are to
2 ! be changed (SYSTEM,OWNER,GROUP,WORLD), while the low word,
2 ! FPROT_VALUE<0,16>, contains the complement of the new protection for each group.
2 ! Thus, if FPROT_VALUE<16,16> is zero, then nothing is to be changed and
2 ! there's no need to go thru the Boolean algebra.
2 !
3 IF .flags[qual_fprot] AND (.fprot_value<16,16> NEQ 0)
2 THEN
2     IF .ods1
2     THEN                            ! For ODS1
3     buffer[hm1$w_fileprot] = (.buffer[hm1$w_fileprot] AND NOT.fprot_value<16,16>)
3                             OR (NOT.fprot_value<0,16> AND .fprot_value<16,16>)
2     ELSE                            ! For ODS2
3     buffer[hm2$w_fileprot] = (.buffer[hm2$w_fileprot] AND NOT.fprot_value<16,16>)
2                             OR (NOT.fprot_value<0,16> AND .fprot_value<16,16>);
2
2 !
```

```
2 ! [NO]HIGHWATER only works for ODS2.
2 !
2 IF .flags[qual_fhw]
2 THEN IF .ods1
2 THEN SIGNAL(set$_notods2, 1, %ASCID 'HIGHWATER_MARKING')
2 ELSE buffer[hm2$v_nohighwater] = .flags[qual_fhw_val];
2
2 !
2 ! In the case of LABEL, the label is stored in the same place on both ODS1 and
2 ! ODS2 disks.  However, there is an additional field in ODS1 homeblocks, which
2 ! contain the volume label, padded with zeroes instead of blanks.
2 !
2 IF .flags[qual_label]
2 THEN
3     BEGIN
3     IF NOT .flags[qual_lbl_cpy]          ! If old label not copied
3     THEN
4         BEGIN                            ! then do so now.
4         CH$MOVE(vcb$s_volname,
4                 buffer[hm1$t_volname2],
4                 label_buff);
4         flags[qual_lbl_cpy] = 1;
3         END;
3     CH$COPY(.label_value[0],             ! Copy label into VOLNAME2,
3             .label_value[1],
3             ' ',                         ! padding with spaces.
3             vcb$s_volname,
3             buffer[hm1$t_volname2]);
3     IF .ods1
3     THEN CH$COPY(.label_value[0],        ! For ODS1, also copy to VOLNAME,
3                 .label_value[1],
3                 0,                       ! padding with zeroes
3                 vcb$s_volname,
3                 buffer[hm1$t_volname]);
2     END;
2
2 !
2 ! For OWNER_UIC, the ODS2 homeblock allows a full 16 bits for group, and
2 ! another 16 bits for member.  In the case of ODS1 disks, each of these fields
2 ! is only 8 bits long.  Also, if fold long UIC's into <377,377> for ODS1 disks.
2
2 IF .flags[qual_owner]
2 THEN
3     BEGIN
3     IF .ods1
3     THEN
4         BEGIN
4         IF .uic_value<8,8> NEQ 0
4         OR .uic_value<24,8> NEQ 0
4         THEN
5             BEGIN
5             uic_value<0,8> = -1;
5             uic_value<8,8> = 0;
5             uic_value<16,8> = -1;
5             uic_value<24,8> = 0;
4             END;
4         buffer[hm1$w_volowner] = (.uic_value<16,8> ^8) + .uic_value<0,8>;
```

```
4            END
3       ELSE buffer[hm2$l_volowner] = .uic_value;
2       END;
2
2   !
2   ! The retention period is something that only exists for ODS2 volumes.  If
2   ! this volume is not an ODS2 disk, then signal an error.  Otherwise, set the
2   ! default retention periods.
2   !
2
2   IF .flags[qual_retent]
2   THEN
3       BEGIN
3       IF .ods1                                ! IF ODS1 disk
3       THEN SIGNAL(set$_notods2,               ! Signal an error
3                   1,
3                   $DESCRIPTOR('/RETENTION'))       ! Saying it can't be done
3       ELSE
4           BEGIN
4           CH$MOVE(8, retmin_value, buffer[hm2$q_retainmin]);
4           CH$MOVE(8, retmax_value, buffer[hm2$q_retainmax]);
3           END;
2       END;
2
2   !
2   ! PROTECTION, the volume protection, is also stored in two different places in
2   ! the home blocks.  See the discussion of the protection value for
2   ! FILE_PROTECTION, above.
2   !
3   IF .flags[qual_vprot] AND (.vprot_value<16,16> NEQ 0)
2   THEN
2       IF .ods1
2       THEN                             ! For ODS1
3       buffer[hm1$w_protect] = (.buffer[hm1$w_protect] AND NOT.vprot_value<16,16>)
3                               OR (NOT.vprot_value<0,16> AND .vprot_value<16,16>)
2       ELSE                             ! For ODS2
3       buffer[hm2$w_protect] = (.buffer[hm2$w_protect] AND NOT.vprot_value<16,16>)
2                               OR (NOT.vprot_value<0,16> AND .vprot_value<16,16>);
2
2
2   !
2   ! WINDOWS is also in two different places.
2   !
2   IF .flags[qual_windows]
2   THEN
3       BEGIN
3       IF .ods1 THEN buffer[hm1$b_window] = .window_value  ! For ODS1
3       ELSE buffer[hm2$b_window] = .window_value;          ! For ODS2
2       END;
2
2   !
2   ! The USER_NAME is in the same place for both types of home blocks.
2   !
2   IF .flags[qual_username]
2   THEN CH$COPY(.user_value[0],            ! Copy the username to the homeblock
2               .user_value[1],
2               ' ',                         ! padded with spaces
; 1555          1544  2                 hm2$s_ownername,
; 1556          1545  2                 buffer[hm2$t_ownername]);
; 1557          1546  2
; 1558          1547  2  !
; 1559          1548  2  ! Recompute the checksums
; 1560          1549  2  !
; 1561          1550  2
; 1562          1551  2  checksum2(buffer, $BYTEOFFSET(hm2$w_checksum1));
; 1563          1552  2  checksum2(buffer, $BYTEOFFSET(hm2$w_checksum2));
; 1564          1553  2
; 1565          1554  2  !
; 1566          1555  2  ! Write the modified homeblock back to the disk
; 1567          1556  2  !
; 1568        P 1557  2  status = $QIOW (CHAN = .channel,
; 1569        P 1558  2                  FUNC = IO$_WRITEVBLK,      ! Read Virtual Block
; 1570        P 1559  2                  IOSB = iosb,
; 1571        P 1560  2                  P1   = buffer,             ! From 'buffer'
; 1572        P 1561  2                  P2   = 512,                ! Write 512 bytes
; 1573          1562  2                  P3   = .vbn);              ! To this virtual block
; 1574          1563  2  IF .status THEN status = .iosb[0];
; 1575          1564  2  IF NOT .status                             ! If not able to read
; 1576          1565  2  THEN
; 1577          1566  3      BEGIN
; 1578          1567  3      SIGNAL(set$_hbwrite,                   ! Error writing a homeblock
; 1579          1568  3             1,
; 1580          1569  3             .desc,                          ! to this disk
; 1581          1570  3             .status);                       ! for this reason
; 1582          1571  3      RETURN false;                          ! don't mod database
; 1583          1572  3      END
; 1584          1573  2  ELSE RETURN true;
; 1585          1574  1  END;


                                                                       .PSECT  $PLIT$,NOWRT,NOEXE,2

                  4B 43 45 48 43 5F 41 54 41 44  0040C P.ADP:  .ASCII  \DATA_CHECK\                          ;
                                                 00416         .BLKB   2
                                     0000000A    00418 P.ADO:  .LONG   10                                    ;
                                     00000000'   0041C         .ADDRESS P.ADP                                ;
45 54 45 4C 45 44 5F 4E 4F 5F 45 53 41 52 45     00420 P.ADR:  .ASCII  \ERASE_ON_DELETE\<0>                   ;
                                           00    0042F                                                       ;
                                     010E000F    00430 P.ADQ:  .LONG   17694735                              ;
                                     00000000'   00434         .ADDRESS P.ADR                                ;
49 4B 52 41 4D 5F 52 45 54 41 57 48 47 49 48     00438 P.ADT:  .ASCII  \HIGHWATER_MARKING\<0><0><0>           ;
                           00 00 00 47 4E        00447                                                       ;
                                     010E0011    0044C P.ADS:  .LONG   17694737                              ;
                                     00000000'   00450         .ADDRESS P.ADT                                ;
                  4E 4F 49 54 4E 45 54 45 52 2F  00454 P.ADV:  .ASCII  \/RETENTION\                          ;
                                                 0045E         .BLKB   2
                                     0000000A    00460 P.ADU:  .LONG   10                                    ;
                                     00000000'   00464         .ADDRESS P.ADV                                ;


                                                                       .PSECT  $CODE$,NOWRT,2
```

```
                                                                  J 4
SETVOL                                                           16-Sep-1984 01:01:55    VAX-11 Bliss-32 V4.0-742                    Page  53
V04-000                                                          14-Sep-1984 12:09:22    [CLIUTL.SRC]SETVOLUME.B32;1                  (12)

                                                        0FFC 00000 SET_HOME:
                                                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11         ; 1316
                                       5B 00000000'   EF 9E  00002         MOVAB   P.ADO, R11                                   ;
                                       5A 00000000G   00 9E  00009         MOVAB   UIC_VALUE, R10                               ;
                                       59 00000000G   00 9E  00010         MOVAB   LIB$SIGNAL, R9                               ;
                                       58 00000000'   EF 9E  00017         MOVAB   FPROT_VALUE, R8                              ;
                                       57 00000000'   EF 9E  0001E         MOVAB   FLAGS, R7                                    ;
                                       5E             08 C2  00025         SUBL2   #8, SP                                       ;
                                                      7E 7C  00028         CLRQ    -(SP)                                        ; 1344
                                                      7E D4  0002A         CLRL    -(SP)                                        ;
                                                 04   AC DD  0002C         PUSHL   VBN                                          ;
                                       7E      0200   8F 3C  0002F         MOVZWL  #512, -(SP)                                  ;
                                                 20   A7 9F  00034         PUSHAB  BUFFER                                       ;
                                                      7E 7C  00037         CLRQ    -(SP)                                        ;
                                                 20   AE 9F  00039         PUSHAB  IOSB                                         ;
                                                      31 DD  0003C         PUSHL   #49                                          ;
                                               0224   C7 DD  0003E         PUSHL   CHANNEL                                      ;
                                                      7E D4  00042         CLRL    -(SP)                                        ;
                             00000000G  00            0C FB  00044         CALLS   #12, SYS$QIOW                                ;
                                       56             50 D0  0004B         MOVL    R0, STATUS                                   ;
                                       06             56 E9  0004E         BLBC    STATUS, 1$                                   ; 1345
                                       56             6E 3C  00051         MOVZWL  IOSB, STATUS                                 ;
                                       10             56 E8  00054         BLBS    STATUS, 2$                                   ; 1346
                                                      56 DD  00057 1$:     PUSHL   STATUS                                       ; 1352
                                                 08   AC DD  00059         PUSHL   DESC                                         ; 1351
                                                      01 DD  0005C         PUSHL   #1                                           ; 1349
                                          00000000G   8F DD  0005E         PUSHL   #SET$_HBREAD                                 ;
                                                    0255 31  00064         BRW     33$                                          ;
                         11            67             01 E1  00067 2$:     BBC     #1, FLAGS, 4$                                ; 1359
                                       07      0221   C7 E9  0006B         BLBC    ODS1, 3$                                     ; 1361
                                  4E   A7        FC   A8 90  00070         MOVB    ACC_VALUE, BUFFER+46                         ;
                                                      05 11  00075         BRB     4$                                           ;
                                  65   A7        FC   A8 90  00077 3$:     MOVB    ACC_VALUE, BUFFER+69                         ; 1362
                         38            67             02 E1  0007C 4$:     BBC     #2, FLAGS, 9$                                ; 1368
                                       0F      0221   C7 E9  00080         BLBC    ODS1, 5$                                     ; 1369
                                                      5B DD  00085         PUSHL   R11                                          ; 1372
                                                      01 DD  00087         PUSHL   #1                                           ; 1370
                                          00000000G   8F DD  00089         PUSHL   #SET$_NOTODS2                                ;
                                       69             03 FB  0008F         CALLS   #3, LIB$SIGNAL                               ;
                                                      24 11  00092         BRB     9$                                           ;
                         04       04   A7             01 E1  00094 5$:     BBC     #1, DFLAGS, 6$                               ; 1375
                                  4A   A7             01 88  00099         BISB2   #1, BUFFER+42                                ;
                         04       04   A7             03 E1  0009D 6$:     BBC     #3, DFLAGS, 7$                               ; 1376
                                  4A   A7             01 8A  000A2         BICB2   #1, BUFFER+42                                ;
                         04       04   A7             02 E1  000A6 7$:     BBC     #2, DFLAGS, 8$                               ; 1377
                                  4A   A7             02 88  000AB         BISB2   #2, BUFFER+42                                ;
                         04       04   A7             04 E1  000AF 8$:     BBC     #4, DFLAGS, 9$                               ; 1378
                                  4A   A7             02 8A  000B4         BICB2   #2, BUFFER+42                                ;
                         21       01   A7             04 E1  000B8 9$:     BBC     #4, FLAGS+1, 11$                             ; 1384
                                       10      0221   C7 E9  000BD         BLBC    ODS1, 10$                                    ; 1385
                                                 18   AB 9F  000C2         PUSHAB  P.ADO                                        ; 1386
                                                      01 DD  000C5         PUSHL   #1                                           ;
                                          00000000G   8F DD  000C7         PUSHL   #SET$_NOTODS2                                ;
                                       69             03 FB  000CD         CALLS   #3, LIB$SIGNAL                               ;
                                                      0C 11  000D0         BRB     11$                                          ;
           50       01   A7            01             05 EF  000D2 10$:    EXTZV   #5, #1, FLAGS+1, R0                          ; 1387
      4A   A7            01            02             50 F0  000D8         INSV    R0, #2, #1, BUFFER+42                        ;
```

```
                 2B          67                03  E1 000DE 11$:   BBC     #3, FLAGS, 14$                              ; 1394
                             52  00000000G     00  D0 000E2        MOVL    EXTE_VALUE, R2                              ; 1398
                             1B       0221     C7  E9 000E9        BLBC    ODS1, 13$                                   ; 1395
                    000000FF 8F                52  D1 000EE        CMPL    R2, #255                                    ; 1398
                                               0C  15 000F5        BLEQ    12$                                         ;
                                  007711EA     8F  DD 000F7        PUSHL   #7803370                                    ; 1401
                             69                01  FB 000FD        CALLS   #1, LIB$SIGNAL                              ;
                                         01C2  31 00100        BRW     35$                                         ; 1402
                          4D A7                52  90 00103 12$:   MOVB    R2, BUFFER+45                               ; 1404
                                               04  11 00107        BRB     14$                                         ; 1395
                          66 A7                52  B0 00109 13$:   MOVW    R2, BUFFER+70                               ; 1406
                 40          67                04  E1 0010D 14$:   BBC     #4, FLAGS, 16$                              ; 1419
                                         02    A8  B5 00111        TSTW    FPROT_VALUE+2                               ;
                                               3B  13 00114        BEQL    16$                                         ;
                             1C       0221     C7  E9 00116        BLBC    ODS1, 15$                                   ; 1421
                             51         44     A7  3C 0011B        MOVZWL  BUFFER+36, R1                               ; 1423
                             50         02     A8  3C 0011F        MOVZWL  FPROT_VALUE+2, R0                           ;
                             51                50  CA 00123        BICL2   R0, R1                                      ;
                             50         02     A8  3C 00126        MOVZWL  FPROT_VALUE+2, R0                           ; 1424
                             52                68  3C 0012A        MOVZWL  FPROT_VALUE, R2                             ;
                             50                52  CA 0012D        BICL2   R2, R0                                      ;
                 44          A7                50                  
                                               51  A9 00130        BISW3   R1, R0, BUFFER+36                           ;
                                               1A  11 00135        BRB     16$                                         ; 1423
                             51         56     A7  3C 00137 15$:   MOVZWL  BUFFER+54, R1                               ; 1426
                             50         02     A8  3C 0013B        MOVZWL  FPROT_VALUE+2, R0                           ;
                             51                50  CA 0013F        BICL2   R0, R1                                      ;
                             50         02     A8  3C 00142        MOVZWL  FPROT_VALUE+2, R0                           ; 1427
                             52                68  3C 00146        MOVZWL  FPROT_VALUE, R2                             ;
                             50                52  CA 00149        BICL2   R2, R0                                      ;
                 56          A7                50                  
                                               51  A9 0014C        BISW3   R1, R0, BUFFER+54                           ;
                 21       01 A7                06  E1 00151 16$:   BBC     #6, FLAGS+1, 18$                            ; 1432
                             10       0221     C7  E9 00156        BLBC    ODS1, 17$                                   ; 1433
                                        34     AB  9F 0015B        PUSHAB  P.ADS                                       ; 1434
                                               01  DD 0015E        PUSHL   #1                                          ;
                                  00000000G    8F  DD 00160        PUSHL   #SET$_NOTODS2                               ;
                             69                03  FB 00166        CALLS   #3, LIB$SIGNAL                              ;
                                               0C  11 00169        BRB     18$                                         ;
         50      01          A7                01  
                                               07  EF 0016B 17$:   EXTZV   #7, #1, FLAGS+1, R0                         ; 1435
   4A    A7                  01                03  
                                               50  F0 00171        INSV    R0, #3, #1, BUFFER+42                       ;
                             29                67  
                                               05  E1 00177 18$:   BBC     #5, FLAGS, 20$                              ; 1442
                             0C         02     A7  06  E0 0017B    BBS     #6, FLAGS+2, 19$                            ; 1445
                 14       A7  01F8             C7  0C  28 00180    MOVC3   #12, BUFFER+472, LABEL_BUFF                 ; 1449
                            02 A7       40     8F  88 00187        BISB2   #64, FLAGS+2                                ; 1451
         0C            20  08 B8        04     A8  2C 0018C 19$:   MOVC5   LABEL_VALUE, @LABEL_VALUE+4, #32, #12, -    ; 1457
                                      01F8     C7     00193                BUFFER+472                                  ;
                             09       0221     C7  E9 00196        BLBC    ODS1, 20$                                   ; 1458
         0C            00  08 B8        04     A8  2C 0019B        MOVC5   LABEL_VALUE, @LABEL_VALUE+4, #0, #12, -     ; 1463
                                        2E     A7     001A2                BUFFER+14                                   ;
                                               67  95 001A4 20$:   TSTB    FLAGS                                       ; 1471
                                               2C  18 001A6        BGEQ    24$                                         ;
                             23       0221     C7  E9 001A8        BLBC    ODS1, 23$                                   ; 1474
                                        01     AA  95 001AD        TSTB    UIC_VALUE+1                                 ; 1477
                                               05  12 001B0        BNEQ    21$                                         ;
                                        03     AA  95 001B2        TSTB    UIC_VALUE+3                                 ; 1478
                                               07  13 001B5        BEQL    22$                                         ;
                             6A  00FF00FF      8F  D0 001B7 21$:   MOVL    #16711935, UIC_VALUE                        ; 1481
                             50         02     AA  9A 001BE 22$:   MOVZBL  UIC_VALUE+2, R0                             ; 1486
```

```
             50               50             08  78 001C2          ASHL    #8, R0, R0                                    ;
                              51             6A  9A 001C6          MOVZBL  UIC_VALUE, R1                                 ;
        3E   A7               50             51  A1 001C9          ADDW3   R1, R0, BUFFER+30                             ;
                                             04  11 001CE          BRB     24$                                           ; 1474
                         4C   A7             6A  D0 001D0 23$:     MOVL    UIC_VALUE, BUFFER+44                          ; 1488
                              21        01   A7  E9 001D4 24$:     BLBC    FLAGS+1, 26$                                  ; 1497
                              10      0221   C7  E9 001D8          BLBC    ODS1, 25$                                     ; 1500
                                        48   AB  9F 001DD          PUSHAB  P.ADU                                         ; 1503
                                             01  DD 001E0          PUSHL   #1                                            ; 1501
                                 00000000G   8F  DD 001E2          PUSHL   #SET$_NOTODS2                                 ;
                                        69   03  FB 001E8          CALLS   #3, LIB$SIGNAL                                ;
                                             0C  11 001EB          BRB     26$                                           ;
        68   A7          10   A8             08  28 001ED 25$:     MOVC3   #8, RETMIN_VALUE, BUFFER+72                   ; 1506
        70   A7          18   A8             08  28 001F3          MOVC3   #8, RETMAX_VALUE, BUFFER+80                   ; 1507
             42          01   A7             02  E1 001F9 26$:     BBC     #2, FLAGS+1, 28$                              ; 1516
                                        0E   A8  B5 001FE          TSTW    VPROT_VALUE+2                                 ;
                                             3D  13 00201          BEQL    28$                                           ;
                              1D      0221   C7  E9 00203          BLBC    ODS1, 27$                                     ; 1518
                              51        40   A7  3C 00208          MOVZWL  BUFFER+32, R1                                 ; 1520
                              50        0E   A8  3C 0020C          MOVZWL  VPROT_VALUE+2, R0                             ;
                              51             50  CA 00210          BICL2   R0, R1                                        ;
                              50        0E   A8  3C 00213          MOVZWL  VPROT_VALUE+2, R0                             ; 1521
                              52        0C   A8  3C 00217          MOVZWL  VPROT_VALUE, R2                               ;
                              50             52  CA 0021B          BICL2   R2, R0                                        ;
        40   A7               50             51  A9 0021E          BISW3   R1, R0, BUFFER+32                             ;
                                             1B  11 00223          BRB     28$                                           ; 1520
                              51        54   A7  3C 00225 27$:     MOVZWL  BUFFER+52, R1                                 ; 1523
                              50        0E   A8  3C 00229          MOVZWL  VPROT_VALUE+2, R0                             ;
                              51             50  CA 0022D          BICL2   R0, R1                                        ;
                              50        0E   A8  3C 00230          MOVZWL  VPROT_VALUE+2, R0                             ; 1524
                              52        0C   A8  3C 00234          MOVZWL  VPROT_VALUE, R2                               ;
                              50             52  CA 00238          BICL2   R2, R0                                        ;
        54   A7               50             51  A9 0023B          BISW3   R1, R0, BUFFER+52                             ;
             11          01   A7             03  E1 00240 28$:     BBC     #3, FLAGS+1, 30$                              ; 1530
                              07      0221   C7  E9 00245          BLBC    ODS1, 29$                                     ; 1533
                         4C   A7        28   A8  90 0024A          MOVB    WINDOW_VALUE, BUFFER+44                       ;
                                             05  11 0024F          BRB     30$                                           ;
                         64   A7        28   A8  90 00251 29$:     MOVB    WINDOW_VALUE, BUFFER+68                       ; 1534
             0A          01   A7             01  E1 00256 30$:     BBC     #1, FLAGS+1, 31$                              ; 1540
  0C         20          24   B8        20   A8  2C 0025B          MOVC5   USER_VALUE, @USER_VALUE+4, #32, #12, -        ; 1545
                                      0204   C7     00262                  BUFFER+484                                    ;
                                             3A  DD 00265 31$:     PUSHL   #58                                           ; 1551
                                        20   A7  9F 00267          PUSHAB  BUFFER                                        ;
                  00000000G   00             02  FB 0026A          CALLS   #2, CHECKSUM2                                 ;
                              7E      01FE   8F  3C 00271          MOVZWL  #510, -(SP)                                   ; 1552
                                        20   A7  9F 00276          PUSHAB  BUFFER                                        ;
                  00000000G   00             02  FB 00279          CALLS   #2, CHECKSUM2                                 ;
                                             7E  7C 00280          CLRQ    -(SP)                                         ; 1562
                                             7E  D4 00282          CLRL    -(SP)                                         ;
                                        04   AC  DD 00284          PUSHL   VBN                                           ;
                              7E      0200   8F  3C 00287          MOVZWL  #512, -(SP)                                   ;
                                        20   A7  9F 0028C          PUSHAB  BUFFER                                        ;
                                             7E  7C 0028F          CLRQ    -(SP)                                         ;
                                        20   AE  9F 00291          PUSHAB  IOSB                                          ;
                                             30  DD 00294          PUSHL   #48                                           ;
                                      0224   C7  DD 00296          PUSHL   CHANNEL                                       ;
                                             7E  D4 0029A          CLRL    -(SP)                                         ;
                  00000000G  00          0C  FB 0029C        CALLS   #12, SYS$QIOW                      ;
                             56          50  D0 002A3        MOVL    R0, STATUS                         ;
                             06          56  E9 002A6        BLBC    STATUS, 32$                        ; 1563
                             56          6E  3C 002A9        MOVZWL  IOSB, STATUS                       ;
                             12          56  E8 002AC        BLBS    STATUS, 34$                        ; 1564
                                         56  DD 002AF  32$:  PUSHL   STATUS                             ; 1570
                                   08    AC  DD 002B1        PUSHL   DESC                               ; 1569
                                         01  DD 002B4        PUSHL   #1                                 ; 1567
                        00000000G        8F  DD 002B6        PUSHL   #SET$_HBWRITE                      ;
                             69          04  FB 002BC  33$:  CALLS   #4, LIB$SIGNAL                     ;
                                         04  11 002BF        BRB     35$                                ; 1573
                             50          01  D0 002C1  34$:  MOVL    #1, R0                             ;
                                             04 002C4        RET                                        ;
                                         50  D4 002C5  35$:  CLRL    R0                                 ; 1574
                                             04 002C7        RET                                        ;

; Routine Size:  712 bytes,    Routine Base:  $CODE$ + 0ADD
```

```
1 ROUTINE set_ucbvcb (ucb) : NOVALUE =
1 !++
1 !
1 ! This routine is called in kernel mode, to modify the fields in the UCB and
1 ! VCB which correspond to changes made in the homeblock.  The address of the
1 ! UCB is passed as the input argument.
1 !
1 !--
2 BEGIN
2
2 MAP ucb : REF $BBLOCK;                                ! Define the UCB
2
2 BIND
2     orb = .ucb[ucb$l_orb] : $BBLOCK,                  ! Define the ORB
2     vcb = .ucb[ucb$l_vcb] : $BBLOCK,                  ! Define the VCB
2     devchar = ucb[ucb$l_devchar] : $BBLOCK;           ! and devchar longword
2
2 !
2 ! Go thru the UCB and VCB, making the same changes to it that were made
2 ! to the homeblock.  Note that, if the LABEL qualifier is set, the volume
2 ! label is changed in the homeblock and in the VCB, but the logical name
2 ! (DISK$label) is NOT CHANGED.
2 !
2
3 IF .flags[qual_access] AND (.acc_inc NEQ 0)
2 THEN vcb[vcb$b_lru_lim] = .vcb[vcb$b_lru_lim] + .acc_inc;
2
3 IF (.flags[qual_data] AND (.buffer[hm2$b_struclev] NEQ 1))
2 THEN
3     BEGIN
3     IF .dflags[data_read] THEN devchar[dev$v_rck] = 1;
3     IF .dflags[data_noread] THEN devchar[dev$v_rck] = 0;
3     IF .dflags[data_write] THEN devchar[dev$v_wck] = 1;
3     IF .dflags[data_nowrite] THEN devchar[dev$v_wck] = 0;
2     END;
2
2 IF .flags[qual_erase]
2 AND NOT .ods1
2 THEN vcb[vcb$v_erase] = .flags[qual_erase_val];
2
2 IF .flags[qual_exte]
2 THEN vcb[vcb$w_extend] = .exte_value;
2
3 IF .flags[qual_fprot] AND (.fprot_value<16,16> NEQ 0)
3 THEN vcb[vcb$w_fileprot] = (.vcb[vcb$w_fileprot] AND NOT .fprot_value<16,16>)
2                           OR (NOT.fprot_value<0,16> AND .fprot_value<16,16>);
2
2 IF .flags[qual_fhw]
2 AND NOT .ods1
2 THEN vcb[vcb$v_nohighwater] = .flags[qual_fhw_val];
2
2 IF .flags[qual_label]
2 THEN CH$COPY(.label_value[0],
2               .label_value[1],
2               ' ',
2               vcb$s_volname,
2               vcb[vcb$t_volname]);
```

```
; 1644          1632  2
; 1645          1633  2  IF .flags[qual_mntver]
; 1646          1634  2  THEN vcb[vcb$v_mountver] = .flags[qual_mntver_val];
; 1647          1635  2
; 1648          1636  2  IF .flags[qual_owner]
; 1649          1637  2  THEN orb[orb$l_owner] = .uic_value;
; 1650          1638  2
; 1651          1639  3  IF .flags[qual_retent] AND (NOT .ods1)
; 1652          1640  2  THEN
; 1653          1641  3      BEGIN
; 1654          1642  3      CH$MOVE(8, retmin_value, vcb[vcb$q_retainmin]);
; 1655          1643  3      CH$MOVE(8, retmax_value, vcb[vcb$q_retainmax]);
; 1656          1644  2      END;
; 1657          1645  2
; 1658          1646  2  IF .flags[qual_unl]
; 1659          1647  2  THEN ucb[ucb$v_unload] = .flags[qual_unl_val];
; 1660          1648  2
; 1661          1649  3  IF .flags[qual_vprot] AND (.vprot_value<16,16> NEQ 0)
; 1662          1650  3  THEN orb[orb$w_prot] = (.orb[orb$w_prot] AND NOT .vprot_value<16,16>)
; 1663          1651  2                          OR (NOT.vprot_value<0,16> AND .vprot_value<16,16>);
; 1664          1652  2  orb[orb$v_prot_16] = 1;                                ! SOGW protection word
; 1665          1653  2
; 1666          1654  2  IF .flags[qual_windows]
; 1667          1655  2  THEN vcb[vcb$b_window] = .window_value;
; 1668          1656  2
; 1669          1657  2  RETURN;
; 1670          1658  1  END;



                                                  07FC 00000 SET_UCBVCB:
                                                                          .WORD    Save R2,R3,R4,R5,R6,R7,R8,R9,R10                 ; 1575
                              5A 00000000'   EF    9E 00002               MOVAB    FPROT_VALUE, R10                                 ;
                              59 00000000'   EF    9E 00009               MOVAB    FLAGS, R9                                        ;
                              57        04   AC    D0 00010               MOVL     UCB, R7                                          ; 1588
                              58        1C   A7    D0 00014               MOVL     28(R7), R8                                       ;
                              56        34   A7    D0 00018               MOVL     52(R7), R6                                       ; 1589
              0C              69             01    E1 0001C               BBC      #1, FLAGS, 1$                                    ; 1599
                                      0220   C9    95 00020               TSTB     ACC_INC                                          ;
                                             06    13 00024               BEQL     1$                                               ;
                        49    A6      0220   C9    80 00026               ADDB2    ACC_INC, 73(R6)                                  ; 1600
              2E              69             02    E1 0002C 1$:           BBC      #2, FLAGS, 5$                                    ; 1602
                              01        2D   A9    91 00030               CMPB     BUFFER+13, #1                                    ;
                                             28    13 00034               BEQL     5$                                               ;
              05        04    A9             01    E1 00036               BBC      #1, DFLAGS, 2$                                   ; 1605
                        3B    A7        40   8F    88 0003B               BISB2    #64, 59(R7)                                      ;
              05        04    A9             03    E1 00040 2$:           BBC      #3, DFLAGS, 3$                                   ; 1606
                        3B    A7        40   8F    8A 00045               BICB2    #64, 59(R7)                                      ;
              05        04    A9             02    E1 0004A 3$:           BBC      #2, DFLAGS, 4$                                   ; 1607
                        3B    A7        80   8F    88 0004F               BISB2    #128, 59(R7)                                     ;
              05        04    A9             04    E1 00054 4$:           BBC      #4, DFLAGS, 5$                                   ; 1608
                        3B    A7        80   8F    8A 00059               BICB2    #128, 59(R7)                                     ;
              11        01    A9             04    E1 0005E 5$:           BBC      #4, FLAGS+1, 6$                                  ; 1611
                              0C      0221   C9    E8 00063               BLBS     ODS1, 6$                                         ; 1612
  50      01  A9                01           05    EF 00068               EXTZV    #5, #1, FLAGS+1, R0                              ; 1613
```

```
   53   A6          01          03                 50 F0 0006E        INSV    R0, #3, #1, 83(R6)                           ;
                    08          69                 03 E1 00074 6$:    BBC     #3, FLAGS, 7$                                ; 1615
                          3E    A6  00000000G      00 B0 00078        MOVW    EXTE_VALUE, 62(R6)                           ; 1616
                    1F          69                 04 E1 00080 7$:    BBC     #4, FLAGS, 8$                                ; 1618
                                           02      AA B5 00084        TSTW    FPROT_VALUE+2                                ;
                                                   1A 13 00087        BEQL    8$                                           ;
                                51         4A      A6 3C 00089        MOVZWL  74(R6), R1                                   ; 1619
                                50         02      AA 3C 0008D        MOVZWL  FPROT_VALUE+2, R0                            ;
                                51                 50 CA 00091        BICL2   R0, R1                                       ;
                                50         02      AA 3C 00094        MOVZWL  FPROT_VALUE+2, R0                            ; 1620
                                52                 6A 3C 00098        MOVZWL  FPROT_VALUE, R2                              ;
                                50                 52 CA 0009B        BICL2   R2, R0                                       ;
                4A  A6          50                 51 A9 0009E        BISW3   R1, R0, 74(R6)                               ;
                    11    01    A9                 06 E1 000A3 8$:    BBC     #6, FLAGS+1, 9$                              ; 1622
                                0C       0221      C9 E8 000A8        BLBS    ODS1, 9$                                     ; 1623
        50      01  A9          01                 07 EF 000AD        EXTZV   #7, #1, FLAGS+1, R0                          ; 1624
   53   A6          01          04                 50 F0 000B3        INSV    R0, #4, #1, 83(R6)                           ;
                    09          69                 05 E1 000B9 9$:    BBC     #5, FLAGS, 10$                               ; 1626
        0C          20    08    BA         04      AA 2C 000BD        MOVC5   LABEL_VALUE, @LABEL_VALUE+4, #32, #12, -     ; 1631
                                           14      A6    000C4                20(R6)                                       ;
                                0C         02      A9 E9 000C6 10$:   BLBC    FLAGS+2, 11$                                 ; 1633
        50      02  A9          01                 01 EF 000CA        EXTZV   #1, #1, FLAGS+2, R0                          ; 1634
   53   A6          01          02                 50 F0 000D0        INSV    R0, #2, #1, 83(R6)                           ;
                                                   69 95 000D6 11$:   TSTB    FLAGS                                        ; 1636
                                                   07 18 000D8        BGEQ    12$                                          ;
                                68  00000000G      00 D0 000DA        MOVL    UIC_VALUE, (R8)                              ; 1637
                                11         01      A9 E9 000E1 12$:   BLBC    FLAGS+1, 13$                                 ; 1639
                                0C       0221      C9 E8 000E5        BLBS    ODS1, 13$                                    ;
                6C  A6    10    AA                 08 28 000EA        MOVC3   #8, RETMIN_VALUE, 108(R6)                    ; 1642
                74  A6    18    AA                 08 28 000F0        MOVC3   #8, RETMAX_VALUE, 116(R6)                    ; 1643
                    0C    02    A9                 02 E1 000F6 13$:   BBC     #2, FLAGS+2, 14$                             ; 1646
        50      02  A9          01                 03 EF 000FB        EXTZV   #3, #1, FLAGS+2, R0                          ; 1647
   65   A7          01          04                 50 F0 00101        INSV    R0, #4, #1, 101(R7)                          ;
                    20    01    A9                 02 E1 00107 14$:   BBC     #2, FLAGS+1, 15$                             ; 1649
                                           0E      AA B5 0010C        TSTW    VPROT_VALUE+2                                ;
                                                   1B 13 0010F        BEQL    15$                                          ;
                                51         18      A8 3C 00111        MOVZWL  24(R8), R1                                   ; 1650
                                50         0E      AA 3C 00115        MOVZWL  VPROT_VALUE+2, R0                            ;
                                51                 50 CA 00119        BICL2   R0, R1                                       ;
                                50         0E      AA 3C 0011C        MOVZWL  VPROT_VALUE+2, R0                            ; 1651
                                52         0C      AA 3C 00120        MOVZWL  VPROT_VALUE, R2                              ;
                                50                 52 CA 00124        BICL2   R2, R0                                       ;
                18  A8          50                 51 A9 00127        BISW3   R1, R0, 24(R8)                               ;
                          0B    A8                 01 88 0012C 15$:   BISB2   #1, 11(R8)                                   ; 1652
                    05    01    A9                 03 E1 00130        BBC     #3, FLAGS+1, 16$                             ; 1654
                          48    A6         28      AA 90 00135        MOVB    WINDOW_VALUE, 72(R6)                         ; 1655
                                                      04 0013A 16$:   RET                                                  ; 1658

; Routine Size:  315 bytes,    Routine Base:  $CODE$ + 0DA5
```

```
  1 ROUTINE modify_volset (desc) : NOVALUE =
  2 BEGIN
  2
  2 !++
  2 !
  2 ! Modify [0,0]VOLSET.SYS on the root volume of the volume set.
  2 ! Only ODS2 initialized volumes can be volume sets so we don't
  2 ! have to worry about the $READ finding the End-of-File value
  2 ! as zero in this case
  2 !
  2 ! Inputs:
  2 !        desc - address of root volume device descriptor
  2 !
  2 ! Outputs:
  2 !        None.
  2 !
  2 !--
  2
  2 MAP
  2     desc : REF VECTOR;
  2
  2 LOCAL
  2     status,
  2     buffer : VECTOR[vsl$c_length,BYTE],
P 2     fab : $FAB(DNM = '[0,0]VOLSET.SYS',
  2               FAC = <get,put,upd>),
P 2     rab : $RAB(FAB = fab,
P 2               UBF = buffer,
  2               USZ = 100);
  2
  2 !
  2 ! Put the root device name in place
  2 !
  2 fab[fab$l_fna] = .desc[1];
  2 fab[fab$b_fns] = .desc[0];
  2
  2 !
  2 ! Open and connect to [0,0]VOLSET.SYS
  2 !
  3 IF (status = $OPEN(FAB = fab))
  2 THEN status = $CONNECT(RAB = rab);
  2 IF NOT .status
  2 THEN
  3     BEGIN
  3     LOCAL
  3         ptr,
  3         d : VECTOR[2],
  3         b : VECTOR[30];
  3     ptr = CH$MOVE(.fab[fab$b_fns],
  3                   .fab[fab$l_fna],
  3                   b);
  3     ptr = CH$MOVE(.fab[fab$b_dns],
  3                   .fab[fab$l_dna],
  3                   .ptr);
  3     SIGNAL(set$_writeerr,
  3            1,
  3            d,
3              .status);
3          END
3
2  ELSE
3      BEGIN
3
3  !
3  ! The first record contains the volume set name.  Skip it.
3  !
3      $GET(RAB = rab);
3
3  !
3  ! Search thru the records until the one matching the saved old label
3  ! is found.  When found, replace the old label with the new one, and
3  ! update the record.
3  !
3      WHILE $GET(RAB = rab) DO
4          BEGIN
4          IF CH$EQL(vcb$s_volname,
4                    label_buff,
4                    vsl$s_name,
4                    buffer,
4                    ' ')
4          THEN
5              BEGIN
5              CH$COPY(.label_value[0],
5                      .label_value[1],
5                      ' ',
5                      vsl$s_name,
5                      buffer);
5              rab[rab$l_rbf] = buffer;
5              rab[rab$w_rsz] = vsl$c_length;
5              $UPDATE(RAB = rab);
5              EXITLOOP
4              END;
3          END;
2      END;
2
2  $CLOSE(FAB = fab);
2
2  RETURN;
1  END;


                                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

53  59  53  2E  54  45  53  4C  4F  56  5D  30  2C  30  5B  00468 P.ADW:  .ASCII  \[0,0]VOLSET.SYS\              ;
                                                            00477         .BLKB   1
                                                        03  00478 P.ADX:  .BYTE   3                               ;
                                                        50  00479         .BYTE   80                              ;
                                                      0000  0047A         .WORD   0                               ;
                                                  00000000  0047C         .LONG   0                               ;
                                                  00000000  00480         .LONG   0                               ;
                                                  00000000  00484         .LONG   0                               ;
                                                  00000000  00488         .LONG   0                               ;
                                                      0000  0048C         .WORD   0                               ;
```

```
      0B  0048E          .BYTE   11
      00  0048F          .BYTE   0
00000000  00490          .LONG   0
      00  00494          .BYTE   0
      00  00495          .BYTE   0
      00  00496          .BYTE   0
      02  00497          .BYTE   2
00000000  00498          .LONG   0
00000000  0049C          .LONG   0
00000000  004A0          .LONG   0
00000000  004A4          .LONG   0
00000000' 004A8          .ADDRESS P.ADW
      00  004AC          .BYTE   0
      0F  004AD          .BYTE   15
    0000  004AE          .WORD   0
00000000  004B0          .LONG   0
    0000  004B4          .WORD   0
      00  004B6          .BYTE   0
      00  004B7          .BYTE   0
00000000  004B8          .LONG   0
00000000  004BC          .LONG   0
    0000  004C0          .WORD   0
      00  004C2          .BYTE   0
      00  004C3          .BYTE   0
00000000  004C4          .LONG   0
      01  004C8 P.ADY:   .BYTE   1
      44  004C9          .BYTE   68
    0000  004CA          .WORD   0
00000000  004CC          .LONG   0
00000000  004D0          .LONG   0
00000000  004D4          .LONG   0
   0000#  004D8          .WORD   0[3]
    0000  004DE          .WORD   0
00000000  004E0          .LONG   0
    0000  004E4          .WORD   0
      00  004E6          .BYTE   0
      00  004E7          .BYTE   0
    0064  004E8          .WORD   100
    0000  004EA          .WORD   0
00000000  004EC          .LONG   0
00000000  004F0          .LONG   0
00000000  004F4          .LONG   0
00000000  004F8          .LONG   0
      00  004FC          .BYTE   0
      00  004FD          .BYTE   0
      00  004FE          .BYTE   0
      00  004FF          .BYTE   0
00000000  00500          .LONG   0
00000000  00504          .LONG   0
00000000  00508          .LONG   0

                         .EXTRN  SYS$CONNECT, SYS$GET
                         .EXTRN  SYS$UPDATE, SYS$CLOSE

                         .PSECT  $CODE$,NOWRT,2

     00FC 00000 MODIFY_VOLSET:
```

```
                                                         .WORD   Save R2,R3,R4,R5,R6,R7                    ; 1659
                      57 00000000G  00     9E 00002      MOVAB   SYS$GET, R7                               ;
                      5E      FEAC  CE     9E 00009      MOVAB   -340(SP), SP                              ;
        FF70   CD 00000000'  EF      0050  8F     28 0000E      MOVC3   #80, P.ADX, FAB                           ; 1684
        0080   CE 00000000'  EF      0044  8F     28 0001A      MOVC3   #68, P.ADY, RAB                           ; 1687
                    00A4  CE        C0  AD     9E 00026      MOVAB   BUFFER, RAB+36                            ; 1684
                    FF68  CD      FF70  CD     9E 0002C      MOVAB   FAB, RAB+60                               ;
                          50        04  AC     D0 00033      MOVL    DESC, R0                                  ; 1692
                 9C   AD            04  A0     D0 00037      MOVL    4(R0), FAB+44                             ;
                 A4   AD                60     90 0003C      MOVB    (R0), FAB+52                              ; 1693
                                  FF70  CD     9F 00040      PUSHAB  FAB                                       ; 1698
           00000000G  00                01     FB 00044      CALLS   #1, SYS$OPEN                              ;
                      56                50     D0 0004B      MOVL    R0, STATUS                                ;
                      11                56     E9 0004E      BLBC    STATUS, 1$                                ;
                                  0080  CE     9F 00051      PUSHAB  RAB                                       ; 1699
           00000000G  00                01     FB 00055      CALLS   #1, SYS$CONNECT                           ;
                      56                50     D0 0005C      MOVL    R0, STATUS                                ;
                      28                56     E8 0005F      BLBS    STATUS, 2$                                ; 1700
                      50            A4  AD     9A 00062 1$:  MOVZBL  FAB+52, R0                                ; 1707
          6E      9C  BD                50     28 00066      MOVC3   R0, @FAB+44, B                            ;
                      50            A5  AD     9A 0006B      MOVZBL  FAB+53, R0                                ; 1710
          63      A0  BD                50     28 0006F      MOVC3   R0, @FAB+48, (PTR)                        ; 1712
                                        56     DD 00074      PUSHL   STATUS                                    ; 1716
                                    7C  AE     9F 00076      PUSHAB  D                                         ; 1713
                                        01     DD 00079      PUSHL   #1                                        ;
                             00000000G  8F     DD 0007B      PUSHL   #SET$_WRITEERR                            ;
           00000000G  00                04     FB 00081      CALLS   #4, LIB$SIGNAL                            ;
                                        42     11 00088      BRB     4$                                        ; 1700
                                  0080  CE     9F 0008A 2$:  PUSHAB  RAB                                       ; 1725
                      67                01     FB 0008E      CALLS   #1, SYS$GET                               ;
                                  0080  CE     9F 00091 3$:  PUSHAB  RAB                                       ; 1732
                      67                01     FB 00095      CALLS   #1, SYS$GET                               ;
                      31                50     E9 00098      BLBC    R0, 4$                                    ;
        C0   AD 00000000'  EF                0C     29 0009B      CMPC3   #12, LABEL_BUFF, BUFFER                   ; 1734
                                        EB     12 000A4      BNEQ    3$                                        ;
 0C          20 00000000'  FF 00000000'  EF     2C 000A6      MOVC5   LABEL_VALUE, @LABEL_VALUE+4, #32, #12, -  ; 1741
                                    C0  AD        000B3              BUFFER                                    ;
                    00A8  CE        C0  AD     9E 000B5      MOVAB   BUFFER, RAB+40                            ; 1746
                    00A2  CE        40  8F     9B 000BB      MOVZBW  #64, RAB+34                               ; 1747
                                  0080  CE     9F 000C1      PUSHAB  RAB                                       ; 1748
           00000000G  00                01     FB 000C5      CALLS   #1, SYS$UPDATE                            ;
                                  FF70  CD     9F 000CC 4$:  PUSHAB  FAB                                       ; 1754
           00000000G  00                01     FB 000D0      CALLS   #1, SYS$CLOSE                             ;
                                               04 000D7      RET                                               ; 1757

; Routine Size:  216 bytes,    Routine Base:  $CODE$ + 0EE0
; 1772          1758  1 GLOBAL ROUTINE COMMON_IO (EFN,CHAN,FUNC,IOSTS,ASTADR,ASTPRM,P1,P2,P3,P4,P5,P6)=
; 1773          1759  1
; 1774          1760  1 !++
; 1775          1761  1 !
; 1776          1762  1 ! FUNCTIONAL DESCRIPTION:
; 1777          1763  1 !
; 1778          1764  1 !       This routine simply executes a $QIOW call with the parameters
; 1779          1765  1 !       supplied. It is called by the MOUNT code that SET links with.
; 1780          1766  1 !
; 1781          1767  1 ! CALLING SEQUENCE:
; 1782          1768  1 !       COMMON_IO (EFN,CHAN,FUNC,IOSTS,ASTADR,ASTPRM,P1,P2,P3,P4,P5,P6)
; 1783          1769  1 !
; 1784          1770  1 ! INPUT PARAMETERS:
; 1785          1771  1 !       As to $QIOW
; 1786          1772  1 !
; 1787          1773  1 ! IMPLICIT INPUTS:
; 1788          1774  1 !       NONE
; 1789          1775  1 !
; 1790          1776  1 ! OUTPUT PARAMETERS:
; 1791          1777  1 !       NONE
; 1792          1778  1 !
; 1793          1779  1 ! IMPLICIT OUTPUTS:
; 1794          1780  1 !       NONE
; 1795          1781  1 !
; 1796          1782  1 ! ROUTINE VALUE:
; 1797          1783  1 !       As to $QIOW
; 1798          1784  1 !
; 1799          1785  1 ! SIDE EFFECTS:
; 1800          1786  1 !       As to $QIOW
; 1801          1787  1 !
; 1802          1788  1 !--
; 1803          1789  1
; 1804          1790  2 BEGIN
; 1805          1791  2
; 1806          1792  2 BUILTIN
; 1807          1793  2         AP,
; 1808          1794  2         CALLG;
; 1809          1795  2
; 1810          1796  2 EXTERNAL ROUTINE
; 1811          1797  2         SYS$QIOW        : ADDRESSING_MODE (GENERAL);
; 1812          1798  2
; 1813          1799  2
; 1814          1800  2 ! We simply pass the call and its parameters along to $QIOW.
; 1815          1801  2 !
; 1816          1802  2
; 1817          1803  2 CALLG (.AP, SYS$QIOW)
; 1818          1804  2
; 1819          1805  1 END;                                    ! End of routine COMMON_IO


                                       0000 00000         .ENTRY  COMMON_IO, Save nothing                          ; 1758
           00000000G  00            6C  FA 00002          CALLG   (AP), SYS$QIOW                                   ; 1803
                                         04 00009          RET                                                      ; 1805
```

```
; Routine Size:  10 bytes,    Routine Base:  $CODE$ + 0FB8
; 1821          1806  1 END
; 1822          1807  0 ELUDOM



                                                          .EXTRN  LIB$SIGNAL, LIB$STOP
;                                 PSECT SUMMARY
;
;       Name                          Bytes                      Attributes
;
;   $GLOBAL$                             48  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $OWN$                               984  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $PLIT$                             1292  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;   $CODE$                             4034  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       181         0        1000           00:01.8
;  _$255$DUA28:[SYSLIB]TPAMAC.L32;1               42         0         0          14           00:00.2



;                                  COMMAND QUALIFIERS

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

; Size:          4034 code + 2324 data bytes
; Run Time:           01:09.3
; Elapsed Time:       03:48.1
; Lines/CPU Min:        1564
; Lexemes/CPU-Min: 23510
; Memory Used:  478 pages
; Compilation Complete
```

0054 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SETPROCES
LIS

SETSHOBRO
LIS

SETVOLUME
LIS

SETPWD
LIS

SETTERM
LIS

SETQUEUE
LIS

SETTIME
LIS

0055 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

SHODEVPRT
LIS

SHODEVUTL
LIS

SHODEV
LIS

SHODEVCLU
LIS