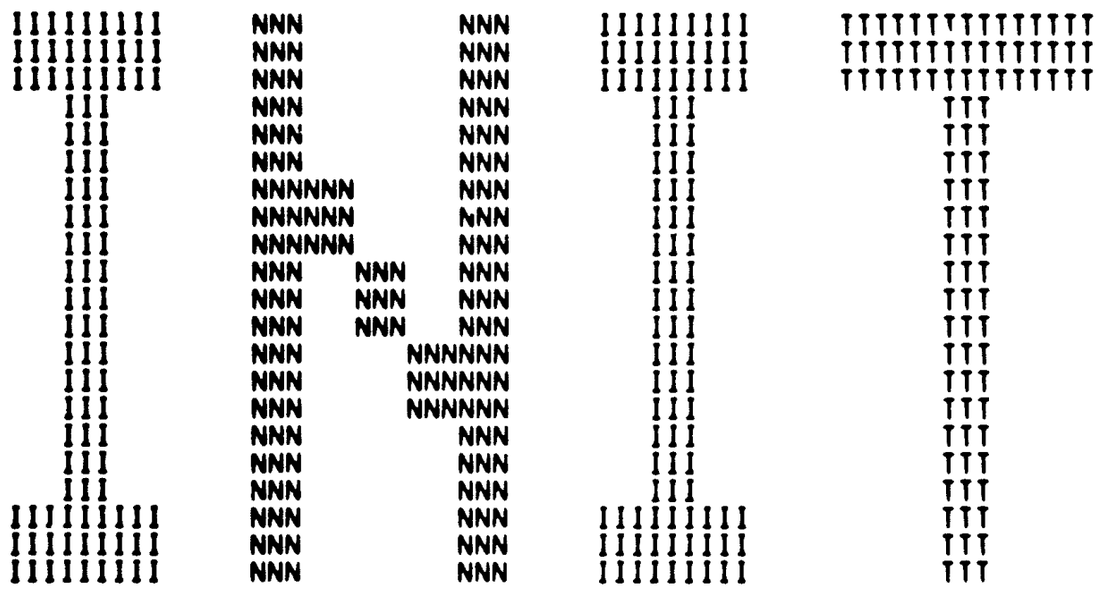
IIIIIIIII NNN NNN IIIIIIIII TTTTTTTTTTTTTTT
IIIIIIIII NNN NNN IIIIIIIII TTTTTTTTTTTTTTT
IIIIIIIII NNN NNN IIIIIIIII TTTTTTTTTTTTTTT
III NNN NNN III TTT
III NNN NNN III TTT
III NNN NNN III TTT
III NNNNNN NNN III TTT
III NNNNNN NNN III TTT
III NNNNNN NNN III TTT
III NNN NNN NNN III TTT
III NNN NNN NNN III TTT
III NNN NNN III TTT
III NNN NNNNNN III TTT
III NNN NNNNNN III TTT
III NNN NNNNNN III TTT
III NNN NNN III TTT
III NNN NNN III TTT
III NNN NNN III TTT
IIIIIIIII NNN NNN IIIIIIIII TTT
IIIIIIIII NNN NNN IIIIIIIII TTT
IIIIIIIII NNN NNN IIIIIIIII TTT

FILEID**INIPAR

```
IIIIII    NN      NN    IIIIII    PPPPPPPP      AAAAAA    RRRRRRRR
IIIIII    NN      NN    IIIIII    PPPPPPPP      AAAAAA    RRRRRRRR
  II      NN      NN      II      PP      PP   AA      AA  RR      RR
  II      NN      NN      II      PP      PP   AA      AA  RR      RR
  II      NNNN    NN      II      PP      PP   AA      AA  RR      RR
  II      NNNN    NN      II      PP      PP   AA      AA  RR      RR
  II      NN  NN  NN      II      PPPPPPPP     AA      AA  RRRRRRRR
  II      NN  NN  NN      II      PPPPPPPP     AA      AA  RRRRRRRR
  II      NN    NNNN      II      PP           AAAAAAAAAA  RR  RR
  II      NN    NNNN      II      PP           AAAAAAAAAA  RR  RR
  II      NN      NN      II      PP           AA      AA  RR    RR      ....
  II      NN      NN      II      PP           AA      AA  RR    RR      ....
IIIIII    NN      NN    IIIIII    PP           AA      AA  RR      RR    ....
IIIIII    NN      NN    IIIIII    PP           AA      AA  RR      RR    ....

LL            IIIIII      SSSSSSSS
LL            IIIIII      SSSSSSSS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II      SS
LL              II        SSSSSS
LL              II        SSSSSS
LL              II              SS
LL              II              SS
LL              II              SS
LL              II              SS
LLLLLLLLLL    IIIIII    SSSSSSSS
LLLLLLLLLL    IIIIII    SSSSSSSS
```

```
;     1          0001  0 MODULE INIPAR (
;     2          0002  0                 LANGUAGE (BLISS32),
;     3          0003  0                 IDENT = 'V04-000'
;     4          0004  0                 ) =
;     5          0005  1 BEGIN
;     6          0006  1
;     7          0007  1 !
;     8          0008  1 !**************************************************************************
;     9          0009  1 !*                                                                        *
;    10          0010  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
;    11          0011  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
;    12          0012  1 !*  ALL RIGHTS RESERVED.                                                   *
;    13          0013  1 !*                                                                        *
;    14          0014  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    15          0015  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    16          0016  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    17          0017  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    18          0018  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    19          0019  1 !*  TRANSFERRED.                                                           *
;    20          0020  1 !*                                                                        *
;    21          0021  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    22          0022  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    23          0023  1 !*  CORPORATION.                                                           *
;    24          0024  1 !*                                                                        *
;    25          0025  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS   *
;    26          0026  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                 *
;    27          0027  1 !*                                                                        *
;    28          0028  1 !*                                                                        *
;    29          0029  1 !**************************************************************************
;    30          0030  1
;    31          0031  1 !++
;    32          0032  1 !
;    33          0033  1 ! FACILITY:  INIT Utility Structure Level 2
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !
;    37          0037  1 !       This module contains the data base and utilities used to acquire the
;    38          0038  1 !       INIT command line from the CLI parser.
;    39          0039  1 !
;    40          0040  1 ! ENVIRONMENT:
;    41          0041  1 !
;    42          0042  1 !       STARLET operating system, including privileged system services
;    43          0043  1 !       and internal exec routines.
;    44          0044  1 !
;    45          0045  1 !--
;    46          0046  1 !
;    47          0047  1 !
;    48          0048  1 ! AUTHOR:  Andrew C. Goldstein,  CREATION DATE:  8-Nov-1977  22:29
;    49          0049  1 !
;    50          0050  1 ! MODIFIED BY:
;    51          0051  1 !
;    52          0052  1 !       V03-005 MCN0140         Maria del C. Nasr       23-Nov-1983
;    53          0053  1 !               Change to new CLI interface.
;    54          0054  1 !
;    55          0055  1 !       V03-004 LMP0140         L. Mark Pilant,         22-Aug-1983  12:51
;    56          0056  1 !               Add support for alphanumeric UICs.
;    57          0057  1 !
```

```
;    58          0058  1 !       V03-003 STJ3089         Steven T. Jeffreys,     24-Apr-1983
;    59          0059  1 !               Add support for /[NO]ERASE and /[NO]HIGHWATER.
;    60          0060  1 !
;    61          0061  1 !       V03-002 MMD0132         Meg Dumont,     12-Apr-1983  17:20
;    62          0062  1 !               Add support for the underscore as a valid character for tape
;    63          0063  1 !
;    64          0064  1 !       V03-001 MMD0128         Meg Dumont,     8-Apr-1983  14:14
;    65          0065  1 !               Added qualifiers to /OVERRIDE and to /LABEL to allow
;    66          0066  1 !               users to ignore and set the OWNER_IDENTIFIER filed
;    67          0067  1 !               of the VOL1 label on ANSI labeled tapes. Added support
;    68          0068  1 !               for the /INTERCHANGE qualifier.
;    69          0069  1 !
;    70          0070  1 !       V02-007 DMW0017         David Michael Walp              19-Jan-1981
;    71          0071  1 !               Uppercase value of volume_accessibility
;    72          0072  1 !
;    73          0073  1 !       V02-006 ACG43021        Andrew C. Goldstein,    4-Jan-1982  16:25
;    74          0074  1 !               Store index file LBN as longword
;    75          0075  1 !
;    76          0076  1 !       V02-005 DMW0015         David Michael Walp              26-Aug-1981
;    77          0077  1 !               /ANSI to /LABEL
;    78          0078  1 !
;    79          0079  1 !       V02-004 DMW0003         David Michael Walp              4-Mar-1981
;    80          0080  1 !               Added /ANSI=VOLUME_ACCESSIBILTY:"x" switch
;    81          0081  1 !
;    82          0082  1 !       V02-003 RLRDENS         Robert L. Rappaport             6-Oct-1980
;    83          0083  1 !               Added /DENSITY=1 and /DENSITY=2 support for RX02's
;    84          0084  1 !
;    85          0085  1 !
;    86          0086  1 !**
;    87          0087  1
;    88          C088  1
;    89          0089  1 LIBRARY 'SYS$LIBRARY:LIB.L32';
;    90          0090  1 REQUIRE 'SRC$:INIDEF.B32';
;    91          0381  1 REQUIRE 'LIBD$:[VMSLIB.OBJ]INITMSG.B32';
;    92          0513  1 LIBRARY 'SYS$LIBRARY:CLIMAC.L32';
;    93          0514  1 LIBRARY 'SYS$LIBRARY:TPAMAC.L32';
```

```
;   95          0515  1 !+
;   96          0516  1 !
;   97          0517  1 ! Impure data area. This area contains the INIT parameters extracted from
;   98          0518  1 ! the command line by the associated parsing routines.
;   99          0519  1 !
;  100          0520  1 !-
;  101          0521  1
;  102          0522  1 LITERAL
;  103          0523  1         BAD_TABLE_LEN   = 100;          ! length of bad block table
;  104          0524  1
;  105          0525  1 GLOBAL
;  106          0526  1         INIT_OPTIONS    : BITVECTOR [64],       ! option flags
;  107          0527  1         PROTECTION,                             ! value of /PROTECTION switch
;  108          0528  1         FILE_PROT,                              ! value of /FILE_PROTECTION switch
;  109          0529  1         MAXIMUM,                                ! value of /MAXIMUM switch
;  110          0530  1         INDEX,                                  ! LBN of index file start
;  111          0531  1         CLUSTER,                                ! value of /CLUSTER switch
;  112          0532  1         HEADERS,                                ! value of /HEADERS switch
;  113          0533  1         DIRECTORIES,                            ! number of MFD entries to preallocate
;  114          0534  1         OWNER_UIC,                              ! value of /OWNER_UIC switch
;  115          0535  1         EXTENSION,                              ! value of /EXTENSION switch
;  116          0536  1         WINDOW,                                 ! value of /WINDOW switch
;  117          0537  1         ACCESSED,                               ! value of /ACCESSED switch
;  118          0538  1         DEVICE_STRING   : BBLOCK [DSC$C_S_BLN], ! decriptor of device name string
;  119          0539  1         LABEL_STRING    : BBLOCK [DSC$C_S_BLN], ! descriptor of volume label string
;  120          0540  1         USER_NAME       : BBLOCK [DSC$C_S_BLN], ! descriptor of user name string
;  121          0541  1         BADBLOCK_COUNT,                         ! count of manually entered bad blocks
;  122          0542  1         BADBLOCK_TABLE  : BBLOCKVECTOR [BAD_TABLE_LEN, BAD_LENGTH],
;  123          0543  1                                                 ! manually entered bad blocks
;  124          0544  1         VOL_ACC         : BYTE,                 ! value of /LABEL:VOLUME_ACCESS switch
;  125          0545  1         VOL_OWNER       : VECTOR [14,BYTE],     ! value of /LABEL=OWNER_ID
;  126          0546  1         DATA_PTR        : VECTOR [32,BYTE],     ! value of OWNER-ID from command line
;  127          0547  1         DATA_INDEX;                             ! index into data_ptr
;  128          0548  1
;  129          0549  1
;  130          0550  1
;  131          0551  1 !
;  132          0552  1 ! Assorted impure data.
;  133          0553  1 !
;  134          0554  1
;  135          0555  1 OWN
;  136          0556  1         CLI_DESC        : BBLOCK [DSC$C_S_BLN],         !CLI work descriptor
;  137          0557  1         TPARSE_BLOCK    . BBLOCK [TPA$K_LENGTH0]
;  138          0558  1                           INITIAL (TPA$R_COUNT0, TPA$M_BLANKS OR TPA$M_ABBREV),
;  139          0559  1         PROT_VAL,
;  140          0560  1         UIC;
;  141          0561  1
;  142          0562  1 EXTERNAL LITERAL
;  143          0563  1         CLI$_DEFAULTED,
;  144          0564  1         CLI$_NEGATED,
;  145          0565  1         CLI$_PRESENT;
;  146          0566  1
```

```
;  148          0567  1 ;
;  149          0568  1 !+
;  150          0569  1 !
;  151          0570  1 ! Request descriptors to the CLI parser. Labels are deemed sufficiently
;  152          0571  1 ! obvious to make cluttering the code with comments unnecessary.
;  153          0572  1 !
;  154          0573  1 !-
;  155          0574  1
;  156          0575  1 BIND
;  157          0576  1         ACCESSED_DESC   = $DESCRIPTOR('ACCESSED'),
;  158          0577  1         BADBLOCKS_DESC  = $DESCRIPTOR('BADBLOCKS'),
;  159          0578  1         CLUSTER_DESC    = $DESCRIPTOR('CLUSTER_SIZE'),
;  160          0579  1         DATA_DESC       = $DESCRIPTOR('DATA_CHECK'),
;  161          0580  1         DENSITY_DESC    = $DESCRIPTOR('DENSITY'),
;  162          0581  1         DIRECT_DESC     = $DESCRIPTOR('DIRECTORIES'),
;  163          0582  1         ERASE_DESC      = $DESCRIPTOR('ERASE'),
;  164          0583  1         EXTENSION_DESC  = $DESCRIPTOR('EXTENSION'),
;  165          0584  1         FILE_DESC       = $DESCRIPTOR('FILE_PROTECTION'),
;  166          0585  1         GROUP_DESC      = $DESCRIPTOR('GROUP'),
;  167          0586  1         HEADERS_DESC    = $DESCRIPTOR('HEADERS'),
;  168          0587  1         HIGH_DESC       = $DESCRIPTOR('HIGHWATER'),
;  169          0588  1         INDEX_DESC      = $DESCRIPTOR('INDEX'),
;  170          0589  1         INTERCHG_DESC   = $DESCRIPTOR('INTERCHANGE'),
;  171          0590  1         LABEL_DESC      = $DESCRIPTOR('LABEL'),
;  172          0591  1         MAXIMUM_DESC    = $DESCRIPTOR('MAXIMUM_FILES'),
;  173          0592  1         OVERRIDE_DESC   = $DESCRIPTOR('OVERRIDE'),
;  174          0593  1         OWNER_DESC      = $DESCRIPTOR('OWNER_UIC'),
;  175          0594  1         PROTECTION_DESC = $DESCRIPTOR('PROTECTION'),
;  176          0595  1         SHARE_DESC      = $DESCRIPTOR('SHARE'),
;  177          0596  1         STRUCTURE_DESC  = $DESCRIPTOR('STRUCTURE'),
;  178          0597  1         SYSTEM_DESC     = $DESCRIPTOR('SYSTEM'),
;  179          0598  1         USER_DESC       = $DESCRIPTOR('USER_NAME'),
;  180          0599  1         VERIFIED_DESC   = $DESCRIPTOR('VERIFIED'),
;  181          0600  1         WINDOW_DESC     = $DESCRIPTOR('WINDOWS');
;  182          0601  1
;  183          0602  1 FORWARD ROUTINE
;  184          0603  1         PARSE_QUALIFIERS: NOVALUE,
;  185          0604  1         DENSITY_ACT     : NOVALUE,
;  186          0605  1         STRUCTURE_ACT   : NOVALUE,
;  187          0606  1         OVERRIDE_ACT    : NOVALUE,
;  188          0607  1         PROTECTION_ACT  : NOVALUE,
;  189          0608  1         FILE_PROT_ACT   : NOVALUE,
;  190          0609  1         OWNER_UIC_ACT   : NOVALUE,
;  191          0610  1         INDEX_ACT       : NOVALUE,
;  192          0611  1         DATACHECK_ACT   : NOVALUE,
;  193          0612  1         USER_NAME_ACT   : NOVALUE,
;  194          0613  1         BADBLOCKS_ACT   : NOVALUE,
;  195          0614  1         LABEL_QUAL_ACT  : NOVALUE;
;  196          0615  1
;  197          0616  1 EXTERNAL ROUTINE
;  198          0617  1         CLI$GET_VALUE,
;  199          0618  1         CLI$PRESENT,
;  200          0619  1         LIB$CVT_DTB;
;  201          0620  1
```

```
;  203          0621  1 GLOBAL ROUTINE INIT_PARSE : NOVALUE =
;  204          0622  1
;  205          0623  1 !++
;  206          0624  1 !
;  207          0625  1 ! FUNCTIONAL DESCRIPTION:
;  208          0626  1 !
;  209          0627  1 !       This routine parses the INIT command line by calling the CLI
;  210          0628  1 !       result parse routines, and leaves the results in the global data
;  211          0629  1 !       area.
;  212          0630  1 !
;  213          0631  1 !
;  214          0632  1 ! CALLING SEQUENCE:
;  215          0633  1 !       INIT_PARSE
;  216          0634  1 !
;  217          0635  1 ! INPUT PARAMETERS:
;  218          0636  1 !
;  219          0637  1 ! IMPLICIT INPUTS:
;  220          0638  1 !       NONE
;  221          0639  1 !
;  222          0640  1 ! OUTPUT PARAMETERS:
;  223          0641  1 !       NONE
;  224          0642  1 !
;  225          0643  1 ! IMPLICIT OUTPUTS:
;  226          0644  1 !       parser impure area on preceding pages
;  227          0645  1 !
;  228          0646  1 ! ROUTINE VALUE:
;  229          0647  1 !       NONE
;  230          0648  1 !
;  231          0649  1 ! SIDE EFFECTS:
;  232          0650  1 !       NONE
;  233          0651  1 !
;  234          0652  1 !--
;  235          0653  1
;  236          0654  2 BEGIN
;  237          0655  2
;  238          0656  2
;  239          0657  2 ! Initialize result parsing.
;  240          0658  2 !
;  241          0659  2
;  242          0660  2 INIT_OPTIONS = INIT_OPTIONS+4 = 0;
;  243          0661  2 INIT_OPTIONS[OPT_NOHIGHWATER] = 1;               ! /NOHIGHWATER leaves it set
;  244          0662  2
;  245          0663  2 ! Initialize descriptor
;  246          0664  2 !
;  247          0665  2 CH$FILL ( 0, DSC$C_S_BLN, CLI_DESC );
;  248          0666  2 CLI_DESC [DSC$B_CLASS] = DSC$K_CLASS_D;
;  249          0667  2
;  250          0668  2
;  251          0669  2 ! Parse command qualifiers. (Most of the action routines are called during
;  252          0670  2 ! this call.)
;  253          0671  2 !
;  254          0672  2 PARSE_QUALIFIERS ();
;  255          0673  2
;  256          0674  2 ! Now acquire device name.
;  257          0675  2 !
;  258          0676  2
;  259          0677  2 CH$FILL ( 0, DSC$C_S_BLN, DEVICE_STRING );
```

```
;  260          0678  2 DEVICE_STRING [DSC$B_CLASS] = DSC$K_CLASS_D;
;  261          0679  2 CLI$GET_VALUE ( $DESCRIPTOR('DEVICE'), DEVICE_STRING );
;  262          0680  2
;  263          0681  2 ! Now acquire volume name.
;  264          0682  2 !
;  265          0683  2
;  266          0684  2 CH$FILL ( 0, DSC$C_S_BLN, LABEL_STRING );
;  267          0685  2 LABEL_STRING [DSC$B_CLASS] = DSC$K_CLASS_D;
;  268          0686  2 CLI$GET_VALUE ( $DESCRIPTOR('VOLUME'), LABEL_STRING );
;  269          0687  2
;  270          0688  2 ! The qualifier /STRUCTURE=1 cannot be used with /CLUSTER or
;  271          0689  2 ! /DATA_CHECK
;  272          0690  2 !
;  273          0691  2
;  274          0692  2 IF .INIT_OPTIONS [OPT_STRUCTURE1]
;  275          0693  3   AND ( .INIT_OPTIONS [OPT_CLUSTER]
;  276          0694  3         OR .INIT_OPTIONS [OPT_READCHECK]
;  277          0695  3         OR .INIT_OPTIONS [OPT_WRITECHECK] )
;  278          0696  2 THEN
;  279          0697  2     ERR_EXIT (INIT$_CONFQUAL);
;  280          0698  2
;  281          0699  1 END;                    ! end of routine INIT_PARSE

                                                        .TITLE  INIPAR
                                                        .IDENT  \V04-000\

                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                    44  45  53  53  45  43  43  41  0000C P.AAB:  .ASCII  \ACCESSED\                    ;
                                    00000008    00008 P.AAA.  .LONG   8                                 ;
                                    00000000'   0000C         .ADDRESS P.AAB                        ;
                53  46  43  4F  4C  42  44  41  42  00010 P.AAD:  .ASCII  \BADBLOCKS\                  ;
                                                00019         .BLKB   3                               ;
                                    00000009    0001C P.AAC:  .LONG   9                               ;
                                    00000000'   00020         .ADDRESS P.AAD                        ;
45  5A  49  53  5F  52  45  54  53  55  4C  43  00024 P.AAF:  .ASCII  \CLUSTER_SIZE\                  ;
                                    0000000C    00030 P.AAE:  .LONG   12                              ;
                                    00000000'   00034         .ADDRESS P.AAF                        ;
        4B  43  45  48  43  5F  41  54  41  44  00038 P.AAH:  .ASCII  \DATA_CHECK\                    ;
                                                00042         .BLKB   2                               ;
                                    0000000A    00044 P.AAG:  .LONG   10                              ;
                                    00000000'   00048         .ADDRESS P.AAH                        ;
                    59  54  49  53  4E  45  44  0004C P.AAJ:  .ASCII  \DENSITY\                     ;
                                                00053         .BLKB   1                               ;
                                    00000007    00054 P.AAI:  .LONG   7                               ;
                                    00000000'   00058         .ADDRESS P.AAJ                        ;
    53  45  49  52  4F  54  43  45  52  49  44  0005C P.AAL:  .ASCII  \DIRECTORIES\                  ;
                                                00067         .BLKB   1                               ;
                                    0000000B    00068 P.AAK:  .LONG   11                              ;
                                    00000000'   0006C         .ADDRESS P.AAL                        ;
                            45  53  41  52  45  00070 P.AAN:  .ASCII  \ERASE\                       ;
                                                00075         .BLKB   3                               ;
                                    00000005    00078 P.AAM:  .LONG   5                               ;
                                    00000000'   0007C         .ADDRESS P.AAN                        ;
                4E  4F  49  53  4E  45  54  58  45  00080 P.AAP:  .ASCII  \EXTENSION\                 ;
                                                00089         .BLKB   3                               ;
```

```
                                00000009  0008C P.AAO:  .LONG   9
                                00000000' 00090         .ADDRESS P.AAP
4E 4F 49 54 43 45 54 4F 52 50 5F 45 4C 49 46  00094 P.AAR:  .ASCII  \FILE_PROTECTION\
                                          000A3         .BLKB   1
                                0000000F  000A4 P.AAQ:  .LONG   15
                                00000000' 000A8         .ADDRESS P.AAR
                           50 55 4F 52 47  000AC P.AAT:  .ASCII  \GROUP\
                                          000B1         .BLKB   3
                                00000005  000B4 P.AAS:  .LONG   5
                                00000000' 000B8         .ADDRESS P.AAT
                        53 52 45 44 41 45 48  000BC P.AAV:  .ASCII  \HEADERS\
                                          000C3         .BLKB   1
                                00000007  000C4 P.AAU:  .LONG   7
                                00000000' 000C8         .ADDRESS P.AAV
                     52 45 54 41 57 48 47 49 48  000CC P.AAX:  .ASCII  \HIGHWATER\
                                          000D5         .BLKB   3
                                00000009  000D8 P.AAW:  .LONG   9
                                00000000' 000DC         .ADDRESS P.AAX
                           58 45 44 4E 49  000E0 P.AAZ:  .ASCII  \INDEX\
                                          000E5         .BLKB   3
                                00000005  000E8 P.AAY:  .LONG   5
                                00000000' 000EC         .ADDRESS P.AAZ
                  45 47 4E 41 48 43 52 45 54 4E 49  000F0 P.ABB:  .ASCII  \INTERCHANGE\
                                          000FB         .BLKB   1
                                0000000B  000FC P.ABA:  .LONG   11
                                00000000' 00100         .ADDRESS P.ABB
                           4C 45 42 41 4C  00104 P.ABD:  .ASCII  \LABEL\
                                          00109         .BLKB   3
                                00000005  0010C P.ABC:  .LONG   5
                                00000000' 00110         .ADDRESS P.ABD
      53 45 4C 49 46 5F 4D 55 4D 49 58 41 4D  00114 P.ABF:  .ASCII  \MAXIMUM_FILES\
                                          00121         .BLKB   3
                                0000000D  00124 P.ABE:  .LONG   13
                                00000000' 00128         .ADDRESS P.ABF
                     45 44 49 52 52 45 56 4F  0012C P.ABH:  .ASCII  \OVERRIDE\
                                00000008  00134 P.ABG:  .LONG   8
                                00000000' 00138         .ADDRESS P.ABH
                  43 49 55 5F 52 45 4E 57 4F  0013C P.ABJ:  .ASCII  \OWNER_UIC\
                                          00145         .BLKB   3
                                00000009  00148 P.ABI:  .LONG   9
                                00000000' 0014C         .ADDRESS P.ABJ
               4E 4F 49 54 43 45 54 4F 52 50  00150 P.ABL:  .ASCII  \PROTECTION\
                                          0015A         .BLKB   2
                                0000000A  0015C P.ABK:  .LONG   10
                                00000000' 00160         .ADDRESS P.ABL
                           45 52 41 48 53  00164 P.ABN:  .ASCII  \SHARE\
                                          00169         .BLKB   3
                                00000005  0016C P.ABM:  .LONG   5
                                00000000' 00170         .ADDRESS P.ABN
                  45 52 55 54 43 55 52 54 53  00174 P.ABP:  .ASCII  \STRUCTURE\
                                          0017D         .BLKB   3
                                00000009  00180 P.ABO:  .LONG   9
                                00000000' 00184         .ADDRESS P.ABP
                           4D 45 54 53 59 53  00188 P.ABR:  .ASCII  \SYSTEM\
                                          0018E         .BLKB   2
                                00000006  00190 P.ABQ:  .LONG   6
                                00000000' 00194         .ADDRESS P.ABR
```

```
                                                00198 P.ABT:    .ASCII   \USER_NAME\                             ;
                                                001A1           .BLKB    3
                                  00000009      001A4 P.ABS:    .LONG    9                                      ;
                                  00000000'     001A8           .ADDRESS P.ABT                                  ;
                                                001AC P.ABV:    .ASCII   \VERIFIED\                              ;
                                  00000008      001B4 P.ABU:    .LONG    8                                      ;
                                  00000000'     001B8           .ADDRESS P.ABV                                  ;
                                                001BC P.ABX:    .ASCII   \WINDOWS\                               ;
                                                001C3           .BLKB    1
                                  00000007      001C4 P.ABW:    .LONG    7                                      ;
                                  00000000'     001C8           .ADDRESS P.ABX                                  ;
                                                001CC P.ABZ:    .ASCII   \DEVICE\                                ;
                                                001D2           .BLKB    2
                                  00000006      001D4 ^.ABY:    .LONG    6                                      ;
                                  00000000'     001D8           .ADDRESS P.ABZ                                  ;
                                                001DC P.ACB:    .ASCII   \VOLUME\                                ;
                                                001E2           .BLKB    2
                                  00000006      001E4 P.ACA:    .LONG    6                                      ;
                                  00000000'     001E8           .ADDRESS P.ACB                                  ;

                                                                .PSECT   $OWN$,NOEXE,2

                                                00000 CLI_DESC:
                                                                .BLKB    8
                        00000003  00000008      00008 TPARSE_BLOCK:
                                                                .LONG    8, 3                                   ;
                                                00010           .BLKB    28
                                                0002C PROT_VAL:
                                                                .BLKB    4
                                                00030 UIC:      .BLKB    4

                                                                .PSECT   $GLOBAL$,NOEXE,2

                                                00000 INIT_OPTIONS::
                                                                .BLKB    8
                                                00008 PROTECTION::
                                                                .BLKB    4
                                                0000C FILE_PROT::
                                                                .BLKB    4
                                                00010 MAXIMUM::
                                                                .BLKB    4
                                                00014 INDEX::   .BLKB    4
                                                00018 CLUSTER::
                                                                .BLKB    4
                                                0001C HEADERS::
                                                                .BLKB    4
                                                00020 DIRECTORIES::
                                                                .BLKB    4
                                                00024 OWNER_UIC::
                                                                .BLKB    4
                                                00028 EXTENSION::
                                                                .BLKB    4
                                                0002C WINDOW::  .BLKB    4
                                                00030 ACCESSED::
                                                                .BLKB    4
                                                00034 DEVICE_STRING::
                                                                .BLKB    8
```

```
0003C LABEL_STRING::
                .BLKB   8
00044 USER_NAME::
                .BLKB   8
0004C BADBLOCK_COUNT::
                .BLKB   4
00050 BADBLOCK_TABLE::
                .BLKB   800
00370 VOL_ACC::
                .BLKB   1
00371           .BLKB   3
00374 VOL_OWNER::
                .BLKB   14
00382           .BLKB   2
00384 DATA_PTR::
                .BLKB   32
003A4 DATA_INDEX::
                .BLKB   4

        ACCESSED_DESC=      P.AAA
        BADBLOCKS_DESC=     P.AAC
        CLUSTER_DESC=       P.AAE
        DATA_DESC=          P.AAG
        DENSITY_DESC=       P.AAI
        DIRECT_DESC=        P.AAK
        ERASE_DESC=         P.AAM
        EXTENSION_DESC=     P.AAO
        FILE_DESC=          P AAQ
        GROUP_DESC=         P.AAS
        HEADERS_DESC=       P.AAU
        HIGH_DESC=          P.AAW
        INDEX_DESC=         P.AAY
        INTERCHG_DESC=      P.ABA
        LABEL_DESC=         P.ABC
        MAXIMUM_DESC=       P.ABE
        OVERRIDE_DESC=      P.ABG
        OWNER_DESC=         P.ABI
        PROTECTION_DESC=    P.ABK
        SHARE_DESC=         P.ABM
        STRUCTURE_DESC=     P.ABO
        SYSTEM_DESC=        P.ABQ
        USER_DESC=          P.ABS
        VERIFIED_DESC=      P.ABU
        WINDOW_DESC=        P.ABW
                .EXTRN  CLI$_DEFAULTED, CLI$_NEGATED
                .EXTRN  CLI$_PRESENT, CLI$GET_VALUE
                .EXTRN  CLI$PRESENT, LIB$CVT_DTB

                .PSECT  $CODE$,NOWRT,2

                007C 00000      .ENTRY  INIT_PARSE, Save R2,R3,R4,R5,R6         ; 0621
        56      0000'  CF   9E  00002   MOVAB   INIT_OPTIONS, R6                ;
                       66   7C  00007   CLRQ    INIT_OPTIONS                    ; 0660
          05    A6             08   88  00009   BISB2   #8, INIT_OPTIONS+5       ; 0661
08        00    6E             00   2C  0000D   MOVC5   #0, (SP), #0, #8, CLI_DESC ; 0665
                       0000'  CF       00012   ;
              0000'    CF             02   90  00015   MOVB    #2, CLI_DESC+3      ; 0666
```

```
                          0000V  CF              00 FB 0001A          CALLS   #0, PARSE_QUALIFIERS                          ; 0672
          08              00     6E              00 2C 0001F          MOVC5   #0, (SP), #0, #8, DEVICE_STRING             ; 0677
                                            34    A6    00024
                            37   A6              02 90 00026          MOVB    #2, DEVICE_STRING+3                         ; 0678
                                            34    A6 9F 0002A          PUSHAB  DEVICE_STRING                               ; 0679
                                       0000'      CF 9F 0002D          PUSHAB  P.ABY
                          0000G  CF               02 FB 00031          CALLS   #2, CLI$GET_VALUE
          08              00     6E              00 2C 00036          MOVC5   #0, (SP), #0, #8, LABEL_STRING              ; 0684
                                            3C    A6    0003B
                            3F   A6              02 90 0003D          MOVB    #2, LABEL_STRING+3                          ; 0685
                                            3C    A6 9F 00041          PUSHAB  LABEL_STRING                                ; 0686
                                       0000'      CF 9F 00044          PUSHAB  P.ACA
                          0000G  CF               02 FB 00048          CALLS   #2, CLI$GET_VALUE
                                            03    A6 95 0004D          TSTB    INIT_OPTIONS+3                              ; 0692
                                                  1A 18 00050          BGEQ    2$
                                            01    A6 95 00052          TSTB    INIT_OPTIONS+1                              ; 0693
                                                  08 19 00055          BLSS    1$
                                                  66 95 00057          TSTB    INIT_OPTIONS                                ; 0694
                                                  04 19 00059          BLSS    1$
                                  0D        01    A6 E9 0005B          BLBC    INIT_OPTIONS+1, 2$                          ; 0695
                                     0075802C   8F DD 0005F 1$:        PUSHL   #7700524                                    ; 0697
                 00000000G        00            01 FB 00065          CALLS   #1, LIB$STOP
                                                  04 0006C 2$:         RET                                                 ; 0699

; Routine Size:  109 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  283          0700  1 ROUTINE PARSE_QUALIFIERS : NOVALUE =
;  284          0701  1
;  285          0702  1 !++
;  286          0703  1 !
;  287          0704  1 ! FUNCTIONAL DESCRIPTION:
;  288          0705  1 !
;  289          0706  1 !       This routine parses the qualifiers of the INIT command line by
;  290          0707  1 !       calling the CLI result parse routines.
;  291          0708  1 !
;  292          0709  1 ! CALLING SEQUENCE:
;  293          0710  1 !       PARSE_QUALIFIERS ()
;  294          0711  1 !
;  295          0712  1 ! INPUT PARAMETERS:
;  296          0713  1 !       NONE
;  297          0714  1 !
;  298          0715  1 ! IMPLICIT INPUTS:
;  299          0716  1 !       NONE
;  300          0717  1 !
;  301          0718  1 ! OUTPUT PARAMETERS:
;  302          0719  1 !       NONE
;  303          0720  1 !
;  304          0721  1 ! IMPLICIT OUTPUTS:
;  305          0722  1 !       INIT_OPTIONS bits set
;  306          0723  1 !
;  307          0724  1 ! ROUTINE VALUE:
;  308          0725  1 !       NONE
;  309          0726  1 !
;  310          0727  1 ! SIDE EFFECTS:
;  311          0728  1 !       NONE
;  312          0729  1 !
;  313          0730  1 !--
;  314          0731  1
;  315          0732  2 BEGIN
;  316          0733  2
;  317          0734  2
;  318          0735  2 ! /ACCESSED qualifier
;  319          0736  2 !
;  320          0737  3 IF ( INIT_OPTIONS [OPT_ACCESSED] = CLI$PRESENT (ACCESSED_DESC) )
;  321          0738  2 THEN
;  322          0739  3     BEGIN
;  323          0740  3     CLI$GET_VALUE ( ACCESSED_DESC, CLI_DESC );
;  324          0741  4     IF NOT ( LIB$CVT_DTB ( .CLI_DESC [DSC$W_LENGTH],
;  325          0742  4                            .CLI_DESC [DSC$A_POINTER],
;  326          0743  4                            ACCESSED ) )
;  327          0744  3     THEN
;  328          0745  3         ERR_EXIT (INIT$_VALCNVERR);
;  329          0746  2     END;
;  330          0747  2
;  331          0748  2 ! /BADBLOCKS qualifier
;  332          0749  2 !
;  333          0750  3 IF ( INIT_OPTIONS [OPT_BADBLOCKS] = CLI$PRESENT (BADBLOCKS_DESC) )
;  334          0751  2 THEN
;  335          0752  2     BADBLOCKS_ACT ();
;  336          0753  2
;  337          0754  2 ! /CLUSTER qualifier
;  338          0755  2 !
;  339          0756  3 IF ( INIT_OPTIONS [OPT_CLUSTER] = CLI$PRESENT (CLUSTER_DESC) )
```

```
;  340          0757  2 THEN
;  341          0758  3     BEGIN
;  342          0759  3     CLI$GET_VALUE ( CLUSTER_DESC, CLI_DESC );
;  343          0760  4     IF NOT ( LIB$CVT_DTB ( .CLI_DESC [DSC$W_LENGTH],
;  344          0761  4                            .CLI_DESC [DSC$A_POINTER],
;  345          0762  4                            CLUSTER ) )
;  346          0763  3     THEN
;  347          0764  3         ERR_EXIT (INIT$_VALCNVERR);
;  348          0765  2     END;
;  349          0766  2
;  350          0767  2 ! /DATA_CHECK qualifier (value not required)
;  351          0768  2 !
;  352          0769  2 IF CLI$PRESENT (DATA_DESC)
;  353          0770  2 THEN
;  354          0771  2     DATACHECK_ACT ();
;  355          0772  2
;  356          0773  2 ! /DENSITY qualifier
;  357          0774  2 !
;  358          0775  3 IF ( INIT_OPTIONS [OPT_DENSITY] = CLI$PRESENT (DENSITY_DESC) )
;  359          0776  2 THEN
;  360          0777  2     DENSITY_ACT ();
;  361          0778  2
;  362          0779  2 ! /DIRECTORIES qualifier
;  363          0780  2 !
;  364          0781  2
;  365          0782  3 IF ( INIT_OPTIONS [OPT_DIRECTORIES] = CLI$PRESENT (DIRECT_DESC) )
;  366          0783  2 THEN
;  367          0784  3     BEGIN
;  368          0785  3     CLI$GET_VALUE ( DIRECT_DESC, CLI_DESC );
;  369          0786  4     IF NOT ( LIB$CVT_DTB ( .CLI_DESC [DSC$W_LENGTH],
;  370          0787  4                            .CLI_DESC [DSC$A_POINTER],
;  371          0788  4                            DIRECTORIES ) )
;  372          0789  3     THEN
;  373          0790  3         ERR_EXIT (INIT$_VALCNVERR);
;  374          0791  2     END;
;  375          0792  2
;  376          0793  2 ! /ERASE qualifier
;  377          0794  2 !
;  378          0795  2 SELECTONE CLI$PRESENT (ERASE_DESC) OF
;  379          0796  2 SET
;  380          0797  2     [CLI$_PRESENT]      : INIT_OPTIONS [OPT_ERASE] = 1;
;  381          0798  2
;  382          0799  2     [CLI$_DEFAULTED,
;  383          0800  2      CLI$_NEGATED]      : INIT_OPTIONS [OPT_ERASE] = 0;
;  384          0801  2 TES;
;  385          0802  2
;  386          0803  2 ! /EXTENSION qualifier
;  387          0804  2 !
;  388          0805  3 IF ( INIT_OPTIONS [OPT_EXTENSION] = CLI$PRESENT (EXTENSION_DESC) )
;  389          0806  2 THEN
;  390          0807  3     BEGIN
;  391          0808  3     CLI$GET_VALUE ( EXTENSION_DESC, CLI_DESC );
;  392          0809  4     IF NOT ( LIB$CVT_DTB ( .CLI_DESC [DSC$W_LENGTH],
;  393          0810  4                            .CLI_DESC [DSC$A_POINTER],
;  394          0811  4                            EXTENSION ) )
;  395          0812  3     THEN
;  396          0813  3         ERR_EXIT (INIT$_VALCNVERR);
```

```
;  397          0814  2      END;
;  398          0815  2
;  399          0816  2  ! /FILE_PROTECTION qualifier
;  400          0817  2  !
;  401          0818  3  IF ( INIT_OPTIONS [OPT_FILE_PROT] = CLI$PRESENT (FILE_DESC) )
;  402          0819  2  THEN
;  403          0820  2      FILE_PROT_ACT ();
;  404          0821  2
;  405          0822  2  ! /GROUP qualifier
;  406          0823  2  !
;  407          0824  2  IF CLI$PRESENT ( GROUP_DESC )
;  408          0825  2  THEN
;  409          0826  2      INIT_OPTIONS [OPT_GROUP] = 1
;  410          0827  2  ELSE
;  411          0828  2      INIT_OPTIONS [OPT_GROUP] = 0;
;  412          0829  2
;  413          0830  2
;  414          0831  2  ! /HEADERS qualifier
;  415          0832  2  !
;  416          0833  3  IF ( INIT_OPTIONS [OPT_HEADERS] = CLI$PRESENT (HEADERS_DESC) )
;  417          0834  2  THEN
;  418          0835  3      BEGIN
;  419          0836  3      CLI$GET_VALUE ( HEADERS_DESC, CLI_DESC );
;  420          0837  4      IF NOT ( LIB$CVT_DTB ( .CLI_DESC [DSC$W_LENGTH],
;  421          0838  4                             .CLI_DESC [DSC$A_POINTER],
;  422          0839  4                             HEADERS ) )
;  423          0840  3      THEN
;  424          0841  3          ERR_EXIT (INIT$_VALCNVERR);
;  425          0842  2      END;
;  426          0843  2
;  427          0844  2  ! /HIGHWATER qualifier
;  428          0845  2  !
;  429          0846  2  SELECTONE CLI$PRESENT (HIGH_DESC) OF
;  430          0847  2  SET
;  431          0848  2      [CLI$_PRESENT,
;  432          0849  2       CLI$_DEFAULTED]    : INIT_OPTIONS [OPT_NOHIGHWATER] = 0;
;  433          0850  2
;  434          0851  2      [CLI$_NEGATED]     : INIT_OPTIONS [OPT_NOHIGHWATER] = 1;
;  435          0852  2  TES;
;  436          0853  2
;  437          0854  2  ! /INDEX qualifier
;  438          0855  2  !
;  439          0856  2  IF CLI$PRESENT (INDEX_DESC)
;  440          0857  2  THEN
;  441          0858  2      INDEX_ACT ();
;  442          0859  2
;  443          0860  2  ! /INTERCHANGE qualifier
;  444          0861  2  !
;  445          0862  2  SELECTONE CLI$PRESENT (INTERCHG_DESC) OF
;  446          0863  2  SET
;  447          0864  2      [CLI$_PRESENT]     : INIT_OPTIONS [OPT_INTERCHG] = 1;
;  448          0865  2
;  449          0866  2      [CLI$_DEFAULTED,
;  450          0867  2       CLI$_NEGATED]     : INIT_OPTIONS [OPT_INTERCHG] = 0;
;  451          0868  2  TES;                                                          ;
;  452          0869  2
;  453          0870  2  ! /LABEL qualifier
```

```
;  454     0871  2 !
;  455     0872  2 IF CLI$PRESENT (LABEL_DESC)
;  456     0873  2 THEN
;  457     0874  2     LABEL_QUAL_ACT ();
;  458     0875  2
;  459     0876  2 ! /MAXIMUM qualifier
;  460     0877  2 !
;  461     0878  3 IF ( INIT_OPTIONS [OPT_MAXIMUM] = CLI$PRESENT (MAXIMUM_DESC) )
;  462     0879  2 THEN
;  463     0880  3     BEGIN
;  464     0881  3     CLI$GET_VALUE ( MAXIMUM_DESC, CLI_DESC );
;  465     0882  4     IF NOT ( LIB$CVT_DTB ( .CLI_DESC [DSC$W_LENGTH],
;  466     0883  4                            .CLI_DESC [DSC$A_POINTER],
;  467     0884  4                            MAXIMUM ) )
;  468     0885  3     THEN
;  469     0886  3         ERR_EXIT (INIT$_VALCNVERR);
;  470     0887  2     END;
;  471     0888  2
;  472     0889  2
;  473     0890  2 ! /OVERRIDE qualifier
;  474     0891  2 !
;  475     0892  2 IF CLI$PRESENT (OVERRIDE_DESC)
;  476     0893  2 THEN
;  477     0894  2     OVERRIDE_ACT ();
;  478     0895  2
;  479     0896  2 ! /OWNER_UIC qualifier
;  480     0897  2 !
;  481     0898  3 IF ( INIT_OPTIONS [OPT_OWNER_UIC] = CLI$PRESENT (OWNER_DESC) )
;  482     0899  2 THEN
;  483     0900  2     OWNER_UIC_ACT ();
;  484     0901  2
;  485     0902  2
;  486     0903  2 ! /PROTECTION qualifier
;  487     0904  2 !
;  488     0905  3 IF ( INIT_OPTIONS [OPT_PROTECTION] = CLI$PRESENT (PROTECTION_DESC) )
;  489     0906  2 THEN
;  490     0907  2     PROTECTION_ACT ();
;  491     0908  2
;  492     0909  2 ! /SHARE qualifier
;  493     0910  2 !
;  494     0911  2 SELECTONE CLI$PRESENT ( SHARE_DESC ) OF
;  495     0912  2 SET
;  496     0913  3     [CLI$_PRESENT]   :  BEGIN
;  497     0914  3                         INIT_OPTIONS [OPT_SHARE] =1;
;  498     0915  3                         INIT_OPTIONS [OPT_EXPNOTMT] = 1;        ! not allowed for tapes
;  499     0916  2                         END;
;  500     0917  2     [CLI$_DEFAULTED] :  INIT_OPTIONS [OPT_SHARE] = 1;
;  501     0918  2     [CLI$_NEGATED]   :  INIT_OPTIONS [OPT_SHARE] = 0;
;  502     0919  2 TES;
;  503     0920  2
;  504     0921  2 ! /STRUCTURE qualifier
;  505     0922  2 !
;  506     0923  2 IF CLI$PRESENT (STRUCTURE_DESC)
;  507     0924  2 THEN
;  508     0925  2     STRUCTURE_ACT ();
;  509     0926  2
;  510     0927  2 ! /SYSTEM qualifier
```

```
;  511          0928  2 !
;  512          0929  2 IF CLI$PRESENT ( SYSTEM_DESC )
;  513          0930  2 THEN
;  514          0931  2     INIT_OPTIONS [OPT_SYSTEM] = 1
;  515          0932  2 ELSE
;  516          0933  2     INIT_OPTIONS [OPT_SYSTEM] = 0;
;  517          0934  2
;  518          0935  2 ! /USER qualifier
;  519          0936  2 !
;  520          0937  3 IF ( INIT_OPTIONS [OPT_USER_NAME] = CLI$PRESENT (USER_DESC) )
;  521          0938  2 THEN
;  522          0939  2     USER_NAME_ACT ();
;  523          0940  2
;  524          0941  2 ! /VERIFIED qualifier
;  525          0942  2 !
;  526          0943  2 SELECTONE CLI$PRESENT ( VERIFIED_DESC ) OF
;  527          0944  2 SET
;  528          0945  3     [CLI$_PRESENT]    :  BEGIN
;  529          0946  3                          INIT_OPTIONS [OPT_VERIFIED] = 1;
;  530          0947  3                          INIT_OPTIONS [OPT_EXP_VER] = 1;
;  531          0948  3                          INIT_OPTIONS [OPT_EXPNOTMT] = 1;        ! not allowed for tapes
;  532          0949  2                          END;
;  533          0950  3     [CLI$_DEFAULTED]  : BEGIN
;  534          0951  3                          INIT_OPTIONS [OPT_VERIFIED] =1;
;  535          0952  3                          INIT_OPTIONS [OPT_EXP_VER] = 0;
;  536          0953  2                          END;
;  537          0954  3     [CLI$_NEGATED]    :  BEGIN
;  538          0955  3                          INIT_OPTIONS [OPT_VERIFIED] = 0;
;  539          0956  3                          INIT_OPTIONS [OPT_EXP_VER] = 1;
;  540          0957  3                          INIT_OPTIONS [OPT_EXPNOTMT] = 1;        ! not allowed for tapes
;  541          0958  2                          END;
;  542          0959  2 TES;
;  543          0960  2
;  544          0961  2 ! /WINDOWS qualifier
;  545          0962  2 !
;  546          0963  3 IF ( INIT_OPTIONS [OPT_WINDOW] = CLI$PRESENT (WINDOW_DESC) )
;  547          0964  2 THEN
;  548          0965  3     BEGIN
;  549          0966  3     CLI$GET_VALUE ( WINDOW_DESC, CLI_DESC );
;  550          0967  3
;  551          0968  4     IF NOT ( LIB$CVT_DTB ( .CLI_DESC [DSC$W_LENGTH],
;  552          0969  4                            .CLI_DESC [DSC$A_POINTER],
;  553          0970  4                            WINDOW ) )
;  554          0971  3     THEN
;  555          0972  3         ERR_EXIT (INIT$_VALCNVERR);
;  556          0973  2     END;
;  557          0974  2
;  558          0975  1 END;                                    ! of PARSE_QUALIFIER routine


                                    0FFC 00000 PARSE_QUALIFIERS:
                                                       .WORD   Save R2,R3,R4,R5,R6,R7,R8,R9,R10,R11        ; 0700
                              5B     0000G  CF  9E 00002       MOVAB   LIB$CVT_DTB, R11                    ;
                              5A     0000G  CF  9E 00007       MOVAB   CLI$GET_VALUE, R10                  ;
```

```
                                59 00000000G   8F   D0 0000C          MOVL    #CLI$_NEGATED, R9                                ;
                                58 00000000G   8F   D0 00013          MOVL    #CLI$_DEFAULTED, R8                              ;
                                57 00000000G   8F   D0 0001A          MOVL    #CLI$_PRESENT, R7                                ;
                                56 00000000G   00   9E 00021          MOVAB   LIB$STOP, R6                                     ;
                                55      0000'  CF   9E 00028          MOVAB   CLI_DESC, R5                                     ;
                                54      0000G  CF   9E 0002D          MOVAB   CLI$PRESENT, R4                                  ;
                                53      0000'  CF   9E 00032          MOVAB   ACCESSED_DESC, R3                                ;
                                52      0000'  CF   9E 00037          MOVAB   INIT_OPTIONS, R2                                 ;
                                               53   DD 0003C          PUSHL   R3                                               ; 0737
                                               64   01   FB 0003E      CALLS   #1, CLI$PRESENT                                  ;
     02   A2            01                     03   50   F0 00041      INSV    R0, #3, #1, INIT_OPTIONS+2                       ;
                                               1D   50   E9 00047      BLBC    R0, 1$                                           ;
                                               28   BB 0004A          PUSHR   #^M<R3,R5>                                       ; 0740
                                               6A   02   FB 0004C      CALLS   #2, CLI$GET_VALUE                                ;
                                                  30   A2   9F 0004F  PUSHAB  ACCESSED                                         ; 0741
                                                  04   A5   DD 00052  PUSHL   CLI_DESC+4                                      ; 0742
                                               7E      65   3C 00055  MOVZWL  CLI_DESC, -(SP)                                 ; 0741
                                               6B      03   FB 00058  CALLS   #3, LIB$CVT_DTB                                 ;
                                               09      50   E8 0005B  BLBS    R0, 1$                                           ;
                                            0075805C   8F   DD 0005E  PUSHL   #7700572                                        ; 0745
                                               66      01   FB 00064  CALLS   #1, LIB$STOP                                    ;
                                                  14   A3   9F 00067 1$:  PUSHAB  BADBLOCKS_DESC                              ; 0750
                                               64      01   FB 0006A  CALLS   #1, CLI$PRESENT                                 ;
     01   A2            01                     01   50   F0 0006D      INSV    R0, #1, #1, INIT_OPTIONS+1                       ;
                                               05   50   E9 00073      BLBC    R0, 2$                                           ;
                                      0000V    CF      00   FB 00076  CALLS   #0, BADBLOCKS_ACT                                ; 0752
                                                  28   A3   9F 0007B 2$:  PUSHAB  CLUSTER_DESC                                ; 0756
                                               64      01   FB 0007E  CALLS   #1, CLI$PRESENT                                 ;
     01   A2            01                     07   50   F0 00081      INSV    R0, #7, #1, INIT_OPTIONS+1                       ;
                                               20   50   E9 00087      BLBC    R0, 3$                                           ;
                                                  55   DD 0008A      PUSHL   R5                                               ; 0759
                                                  28   A3   9F 0008C  PUSHAB  CLUSTER_DESC                                    ;
                                               6A      02   FB 0008F  CALLS   #2, CLI$GET_VALUE                               ;
                                                  18   A2   9F 00092  PUSHAB  CLUSTER                                         ; 0760
                                                  04   A5   DD 00095  PUSHL   CLI_DESC+4                                      ; 0761
                                               7E      65   3C 00098  MOVZWL  CLI_DESC, -(SP)                                 ; 0760
                                               6B      03   FB 0009B  CALLS   #3, LIB$CVT_DTB                                 ;
                                               09      50   E8 0009E  BLBS    R0, 3$                                           ;
                                            0075805C   8F   DD 000A1  PUSHL   #7700572                                        ; 0764
                                               66      01   FB 000A7  CALLS   #1, LIB$STOP                                    ;
                                                  3C   A3   9F 000AA 3$:  PUSHAB  DATA_DESC                                   ; 0769
                                               64      01   FB 000AD  CALLS   #1, CLI$PRESENT                                 ;
                                               05      50   E9 000B0  BLBC    R0, 4$                                           ;
                                      0000V    CF      00   FB 000B3  CALLS   #0, DATACHECK_ACT                                ; 0771
                                                  4C   A3   9F 000B8 4$:  PUSHAB  DENSITY_DESC                                ; 0775
                                               64      01   FB 000BB  CALLS   #1, CLI$PRESENT                                 ;
          62       01                        00      50   F0 000BE      INSV    R0, #0, #1, INIT_OPTIONS                         ;
                                               05   50   E9 000C3      BLBC    R0, 5$                                           ;
                                      0000V    CF      00   FB 000C6  CALLS   #0, DENSITY_ACT                                  ; 0777
                                                  60   A3   9F 000CB 5$:  PUSHAB  DIRECT_DESC                                  ; 0782
                                               64      01   FB 000CE  CALLS   #1, CLI$PRESENT                                 ;
     03   A2            01                     05   50   F0 000D1      INSV    R0, #5, #1, INIT_OPTIONS+3                       ;
                                               20   50   E9 000D7      BLBC    R0, 6$                                           ;
                                                  55   DD 000DA      PUSHL   R5                                               ; 0785
                                                  60   A3   9F 000DC  PUSHAB  DIRECT_DESC                                      ;
                                               6A      02   FB 000DF  CALLS   #2, CLI$GET_VALUE                               ;
                                                  20   A2   9F 000E2  PUSHAB  DIRECTORIES                                      ; 0786
```

```
                              04   A5  DD 000E5       PUSHL   CLI_DESC+4                              ; 0787
                      7E           65  3C 000E8       MOVZWL  CLI_DESC, -(SP)                         ; 0786
                      6B           03  FB 000EB       CALLS   #3, LIB$CVT_DTB                         ;
                      09           50  E8 000EE       BLBS    R0, 6$                                  ;
                        0075805C   8F  DD 000F1       PUSHL   #7700572                                ; 0790
                      66           01  FB 000F7       CALLS   #1, LIB$STOP                            ;
                              70   A3  9F 000FA 6$:   PUSHAB  ERASE_DESC                              ; 0795
                      64           01  FB 000FD       CALLS   #1, CLI$PRESENT                         ;
                      57           50  D1 00100       CMPL    R0, R7                                  ; 0797
                                   06  12 00103       BNEQ    7$                                      ;
                 05   A2           04  88 00105       BISB2   #4, INIT_OPTIONS+5                      ;
                                   0E  11 00109       BRB     9$                                      ;
                      58           50  D1 0010B 7$:   CMPL    R0, R8                                  ; 0799
                                   05  13 0010E       BEQL    8$                                      ;
                      59           50  D1 00110       CMPL    R0, R9                                  ;
                                   04  12 00113       BNEQ    9$                                      ;
                 05   A2           04  8A 00115 8$:   BICB2   #4, INIT_OPTIONS+5                      ; 0800
                            0084   C3  9F 00119 9$:   PUSHAB  EXTENSION_DESC                          ; 0805
                      64           01  FB 0011D       CALLS   #1, CLI$PRESENT                         ;
  02   A2        01   01           50  F0 00120       INSV    R0, #1, #1, INIT_OPTIONS+2              ;
                      21           50  E9 00126       BLBC    R0, 10$                                 ;
                                   55  DD 00129       PUSHL   R5                                      ; 0808
                            0084   C3  9F 0012B       PUSHAB  EXTENSION_DESC                          ;
                      6A           02  FB 0012F       CALLS   #2, CLI$GET_VALUE                       ;
                              28   A2  9F 00132       PUSHAB  EXTENSION                               ; 0809
                              04   A5  DD 00135       PUSHL   CLI_DESC+4                              ; 0810
                      7E           65  3C 00138       MOVZWL  CLI_DESC, -(SP)                         ; 0809
                      6B           03  FB 0013B       CALLS   #3, LIB$CVT_DTB                         ;
                      09           50  E8 0013E       BLBS    R0, 10$                                 ;
                        0075805C   8F  DD 00141       PUSHL   #7700572                                ; 0813
                      66           01  FB 00147       CALLS   #1, LIB$STOP                            ;
                            009C   C3  9F 0014A 10$:  PUSHAB  FILE_DESC                               ; 0818
                      64           01  FB 0014E       CALLS   #1, CLI$PRESENT                         ;
  01   A2        01   03           50  F0 00151       INSV    R0, #3, #1, INIT_OPTIONS+1              ;
                      05           50  E9 00157       BLBC    R0, 11$                                 ;
               0000V  CF           00  FB 0015A       CALLS   #0, FILE_PROT_ACT                       ; 0820
                            00AC   C3  9F 0015F 11$:  PUSHAB  GROUP_DESC                              ; 0824
                      64           01  FB 00163       CALLS   #1, CLI$PRESENT                         ;
                      05           50  E9 00166       BLBC    R0, 12$                                 ;
                      62           08  88 00169       BISB2   #8, INIT_OPTIONS                        ; 0826
                                   03  11 0016C       BRB     13$                                     ;
                      62           08  8A 0016E 12$:  BICB2   #8, INIT_OPTIONS                        ; 0828
                            00BC   C3  9F 00171 13$:  PUSHAB  HEADERS_DESC                            ; 0833
                      64           01  FB 00175       CALLS   #1, CLI$PRESENT                         ;
  02   A2        01   00           50  F0 00178       INSV    R0, #0, #1, INIT_OPTIONS+2              ;
                      21           50  E9 0017E       BLBC    R0, 14$                                 ;
                                   55  DD 00181       PUSHL   R5                                      ; 0836
                            00BC   C3  9F 00183       PUSHAB  HEADERS_DESC                            ;
                      6A           02  FB 00187       CALLS   #2, CLI$GET_VALUE                       ;
                              1C   A2  9F 0018A       PUSHAB  HEADERS                                 ; 0837
                              04   A5  DD 0018D       PUSHL   CLI_DESC+4                              ; 0838
                      7E           65  3C 00190       MOVZWL  CLI_DESC, -(SP)                         ; 0837
                      6B           03  FB 00193       CALLS   #3, LIB$CVT_DTB                         ;
                      09           50  E8 00196       BLBS    R0, 14$                                 ;
                        0075805C   8F  DD 00199       PUSHL   #7700572                                ; 0841
                      66           01  FB 0019F       CALLS   #1, LIB$STOP                            ;
                            00D0   C3  9F 001A2 14$:  PUSHAB  HIGH_DESC                               ; 0846
```

```
                        64              01 FB 001A6          CALLS   #1, CLI$PRESENT
                        57              50 D1 001A9          CMPL    R0, R7                                  ;0848
                                        05 13 001AC          BEQL    15$
                        58              50 D1 001AE          CMPL    R0, R8
                                        06 12 001B1          BNEQ    16$
                    05  A2              08 8A 001B3 15$:     BICB2   #8, INIT_OPTIONS+5                      ;0849
                                        09 11 001B7          BRB     17$
                        59              50 D1 001B9 16$:     CMPL    R0, R9                                  ;0851
                                        04 12 001BC          BNEQ    17$
                    05  A2              08 88 001BE          BISB2   #8, INIT_OPTIONS+5
                          00E0      C3  9F 001C2 17$:     PUSHAB  INDEX_DESC                              ;0856
                        64              01 FB 001C6          CALLS   #1, CLI$PRESENT
                        05              50 E9 001C9          BLBC    R0, 18$
                0000V   CF              00 FB 001CC          CALLS   #0, INDEX_ACT                           ;0858
                          00F4      C3  9F 001D1 18$:     PUSHAB  INTERCHG_DESC                           ;0862
                        64              01 FB 001D5          CALLS   #1, CLI$PRESENT
                        57              50 D1 001D8          CMPL    R0, R7                                  ;0864
                                        06 12 001DB          BNEQ    19$
                    05  A2              02 88 001DD          BISB2   #2, INIT_OPTIONS+5
                                        0E 11 001E1          BRB     21$
                        58              50 D1 001E3 19$:     CMPL    R0, R8                                  ;0866
                                        05 13 001E6          BEQL    20$
                        59              50 D1 001E8          CMPL    R0, R9
                                        04 12 001EB          BNEQ    21$
                    05  A2              02 8A 001ED 20$:     BICB2   #2, INIT_OPTIONS+5                      ;0867
                          0104      C3  9F 001F1 21$:     PUSHAB  LABEL_DESC                              ;0872
                        64              01 FB 001F5          CALLS   #1, CLI$PRESENT
                        05              50 E9 001F8          BLBC    R0, 22$
                0000V   CF              00 FB 001FB          CALLS   #0, LABEL_QUAL_ACT                      ;0874
                          011C      C3  9F 00200 22$:     PUSHAB  MAXIMUM_DESC                            ;0878
                        64              01 FB 00204          CALLS   #1, CLI$PRESENT
01  A2              01  06              50 F0 00207          INSV    R0, #6, #1, INIT_OPTIONS+1
                        21              50 E9 0020D          BLBC    R0, 23$
                                        55 DD 00210          PUSHL   R5                                      ;0881
                          011C      C3  9F 00212          PUSHAB  MAXIMUM_DESC
                        6A              02 FB 00216          CALLS   #2, CLI$GET_VALUE
                            10      A2  9F 00219          PUSHAB  MAXIMUM                                 ;0882
                            04      A5  DD 0021C          PUSHL   CLI_DESC+4                              ;0883
                        7E              65 3C 0021F          MOVZWL  CLI_DESC, -(SP)                         ;0882
                        6B              03 FB 00222          CALLS   #3, LIB$CVT_DTB
                        09              50 E8 00225          BLBS    R0, 23$
                            0075805C    8F DD 00228          PUSHL   #7700572                                ;0886
                        66              01 FB 0022E          CALLS   #1, LIB$STOP
                          012C      C3  9F 00231 23$:     PUSHAB  OVERRIDE_DESC                           ;0892
                        64              01 FB 00235          CALLS   #1, CLI$PRESENT
                        05              50 E9 00238          BLBC    R0, 24$
                0000V   CF              00 FB 0023B          CALLS   #0, OVERRIDE_ACT                        ;0894
                          0140      C3  9F 00240 24$:     PUSHAB  OWNER_DESC                              ;0898
                        64              01 FB 00244          CALLS   #1, CLI$PRESENT
01  A2              01  05              50 F0 00247          INSV    R0, #5, #1, INIT_OPTIONS+1
                        05              50 E9 0024D          BLBC    R0, 25$
                0000V   CF              00 FB 00250          CALLS   #0, OWNER_UIC_ACT                       ;0900
                          0154      C3  9F 00255 25$:     PUSHAB  PROTECTION_DESC                         ;0905
                        64              01 FB 00259          CALLS   #1, CLI$PRESENT
01  A2              01  02              50 F0 0025C          INSV    R0, #2, #1, INIT_OPTIONS+1
                        05              50 E9 00262          BLBC    R0, 26$
                0000V   CF              00 FB 00265          CALLS   #0, PROTECTION_ACT                      ;0907
```

```
                                                0164  C3  9F 0026A 26$:    PUSHAB  SHARE_DESC                              ; 0911
                                                64    01  FB 0026E         CALLS   #1, CLI$PRESENT                         ;
                                                57    50  D1 00271         CMPL    R0, R7                                  ; 0913
                                                      09  12 00274         BNEQ    27$                                     ;
                                                62    04  88 00276         BISB2   #4, INIT_OPTIONS                        ; 0914
                                          04    A2    01  88 00279         BISB2   #1, INIT_OPTIONS+4                      ; 0915
                                                      12  11 0027D         BRB     29$                                     ; 0911
                                                58    50  D1 0027F 27$:    CMPL    R0, R8                                  ; 0917
                                                      05  12 00282         BNEQ    28$                                     ;
                                                62    04  88 00284         BISB2   #4, INIT_OPTIONS                        ;
                                                      08  11 00287         BRB     29$                                     ;
                                                59    50  D1 00289 28$:    CMPL    R0, R9                                  ; 0918
                                                      03  12 0028C         BNEQ    29$                                     ;
                                                62    04  8A 0028E         BICB2   #4, INIT_OPTIONS                        ;
                                                0178  C3  9F 00291 29$:    PUSHAB  STRUCTURE_DESC                          ; 0923
                                                64    01  FB 00295         CALLS   #1, CLI$PRESENT                         ;
                                                05    50  E9 00298         BLBC    R0, 30$                                 ;
                                          0000V CF    00  FB 0029B         CALLS   #0, STRUCTURE_ACT                       ; 0925
                                                0188  C3  9F 002A0 30$:    PUSHAB  SYSTEM_DESC                             ; 0929
                                                64    01  FB 002A4         CALLS   #1, CLI$PRESENT                         ;
                                                05    50  E9 002A7         BLBC    R0, 31$                                 ;
                                                62    10  88 002AA         BISB2   #16, INIT_OPTIONS                       ; 0931
                                                      03  11 002AD         BRB     32$                                     ;
                                                62    10  8A 002AF 31$:    BICB2   #16, INIT_OPTIONS                       ; 0933
                                                019C  C3  9F 002B2 32$:    PUSHAB  USER_DESC                               ; 0937
                                                64    01  FB 002B6         CALLS   #1, CLI$PRESENT                         ;
03    A2        01                              04    50  F0 002B9         INSV    R0, #4, #1, INIT_OPTIONS+3              ;
                                                05    50  E9 002BF         BLBC    R0, 33$                                 ;
                                          0000V CF    00  FB 002C2         CALLS   #0, USER_NAME_ACT                       ; 0939
                                                01AC  C3  9F 002C7 33$:    PUSHAB  VERIFIED_DESC                           ; 0943
                                                64    01  FB 002CB         CALLS   #1, CLI$PRESENT                         ;
                                                57    50  D1 002CE         CMPL    R0, R7                                  ; 0945
                                                      06  12 002D1         BNEQ    34$                                     ;
                                                62    40  8F 88 002D3         BISB2   #64, INIT_OPTIONS                       ; 0946
                                                      18  11 002D7         BRB     36$                                     ; 0947
                                                58    50  D1 002D9 34$:    CMPL    R0, R8                                  ; 0950
                                                      0A  12 002DC         BNEQ    35$                                     ;
                                                62    40  8F 88 002DE         BISB2   #64, INIT_OPTIONS                       ; 0951
                                          04    A2    04  8A 002E2         BICB2   #4, INIT_OPTIONS+4                      ; 0952
                                                      0D  11 002E6         BRB     37$                                     ; 0943
                                                59    50  D1 002E8 35$:    CMPL    R0, R9                                  ; 0954
                                                      08  12 002EB         BNEQ    37$                                     ;
                                                62    40  8F 8A 002ED         BICB2   #64, INIT_OPTIONS                       ;
                                          04    A2    05  88 002F1 36$:    BISB2   #5, INIT_OPTIONS+4                      ; 0957
                                                01BC  C3  9F 002F5 37$:    PUSHAB  WINDOW_DESC                             ; 0963
                                                64    01  FB 002F9         CALLS   #1, CLI$PRESENT                         ;
02    A2        01                              02    50  F0 002FC         INSV    R0, #2, #1, INIT_OPTIONS+2              ;
                                                21    50  E9 00302         BLBC    R0, 38$                                 ;
                                                      55  DD 00305         PUSHL   R5                                      ; 0966
                                                01BC  C3  9F 00307         PUSHAB  WINDOW_DESC                             ;
                                                6A    02  FB 0030B         CALLS   #2, CLI$GET_VALUE                       ;
                                                      2C  A2 9F 0030E         PUSHAB  WINDOW                                  ; 0968
                                                      04  A5 DD 00311         PUSHL   CLI_DESC+4                              ; 0969
                                                7E          65  3C 00314         MOVZWL  CLI_DESC, -(SP)                         ; 0968
                                                6B          03  FB 00317         CALLS   #3, LIB$CVT_DTB                         ;
                                                09          50  E8 0031A         BLBS    R0, 38$                                 ;
                                                      0075805C  8F  DD 0031D         PUSHL   #7700572                                ; 0972
```

```
                    66          01  FB 00323                CALLS   #1, LIB$STOP                        ;
                                    04 00326 38$:           RET                                         ; 0975

; Routine Size:  807 bytes,    Routine Base:  $CODE$ + 006D
```

```
;  560          0976  1 !+
;  561          0977  1 !
;  562          0978  1 ! Parameter and qualifier action routines. Each routine is named corresponding
;  563          0979  1 ! to its associated parameter of qualifier. Each routine does whatever
;  564          0980  1 ! conversion is necessary and stores the parameter or qualifier value in
;  565          0981  1 ! the appropriate location in the output area.
;  566          0982  1 !
;  567          0983  1 !-
;  568          0984  1
;  569          0985  1
;  570          0986  1 ROUTINE BADBLOCKS_ACT (REQ_DESC, CLI_CALLBACK) : NOVALUE =
;  571          0987  2 BEGIN
;  572          0988  2
;  573          0989  2 EXTERNAL
;  574          0990  2         BADBLOCKS_STB   : VECTOR [0],   ! state table address
;  575          0991  2         BADBLOCKS_KTB   : VECTOR [0];   ! keyword table address
;  576          0992  2
;  577          0993  2 EXTERNAL ROUTINE
;  578          0994  2         LIB$TPARSE;
;  579          0995  2
;  580          0996  2 ! Parse the BADBLOCKS options string.
;  581          0997  2 !
;  582          0998  2
;  583          0999  2 WHILE CLI$GET_VALUE ( BADBLOCKS_DESC, CLI_DESC ) DO
;  584          1000  3 BEGIN
;  585          1001  3     TPARSE_BLOCK[TPA$L_STRINGCNT] = .CLI_DESC [DSC$W_LENGTH];
;  586          1002  3     TPARSE_BLOCK[TPA$L_STRINGPTR] = .CLI_DESC [DSC$A_POINTER];
;  587          1003  3     IF NOT LIB$TPARSE (TPARSE_BLOCK, BADBLOCKS_STB, BADBLOCKS_KTB)
;  588          1004  3     THEN
;  589          1005  3         ERR_EXIT (INIT$_BADBLOCKS);
;  590          1006  2 END;
;  591          1007  2
;  592          1008  1 END;                                    ! end of routine BADBLOCKS_ACT


                                                        .EXTRN  BADBLOCKS_STB, BADBLOCKS_KTB
                                                        .EXTRN  LIB$TPARSE

                                        0004 00000 BADBLOCKS_ACT:
                                                        .WORD   Save R2                                 ; 0986
                        52      0000'   CF  9E 00002            MOVAB   CLI_DESC, R2                    ;
                                        52  DD 00007 1$:      PUSHL   R2                              ; 0999
                                0000'   CF  9F 00009            PUSHAB  BADBLOCKS_DESC                  ;
                  0000G   CF            02  FB 0000D            CALLS   #2, CLI$GET_VALUE               ;
                          2B            50  E9 00012            BLBC    R0, 2$                          ;
                    10    A2            62  3C 00015            MOVZWL  CLI_DESC, TPARSE_BLOCK+8        ; 1001
                    14    A2      04    A2  D0 00019            MOVL    CLI_DESC+4, TPARSE_BLOCK+12     ; 1002
                                0000G   CF  9F 0001E            PUSHAB  BADBLOCKS_KTB                   ; 1003
                                0000G   CF  9F 00022            PUSHAB  BADBLOCKS_STB                   ;
                                  08    A2  9F 00026            PUSHAB  TPARSE_BLOCK                    ;
                  0000G   CF            03  FB 00029            CALLS   #3, LIB$TPARSE                  ;
                          D6            50  E8 0002E            BLBS    R0, 1$                          ;
                              00758084  8F  DD 00031            PUSHL   #7700612                        ; 1005
              00000000G   00            01  FB 00037            CALLS   #1, LIB$STOP                    ;
                                        C7  11 0003E            BRB     1$                              ; 0999
                                            04 00040 2$:        RET                                     ; 1008
```

; Routine Size: 65 bytes,   Routine Base: $CODE$ + 0394

```
;  594          1009  1 ROUTINE DATACHECK_ACT : NOVALUE =
;  595          1010  2 BEGIN
;  596          1011  2
;  597          1012  2 EXTERNAL
;  598          1013  2         DATACHECK_STB   : VECTOR [0],   ! state table address
;  599          1014  2         DATACHECK_KTB   : VECTOR [0];   ! keyword table address
;  600          1015  2
;  601          1016  2 EXTERNAL ROUTINE
;  602          1017  2         LIB$TPARSE;
;  603          1018  2
;  604          1019  2 LOCAL
;  605          1020  2     VALUE_FOUND;                        ! set when value present
;  606          1021  2
;  607          1022  2 ! Parse the DATACHECK options string.
;  608          1023  2 !
;  609          1024  2
;  610          1025  2 VALUE_FOUND = 0;
;  611          1026  2
;  612          1027  2 WHILE CLI$GET_VALUE (DATA_DESC, CLI_DESC) DO
;  613          1028  3 BEGIN
;  614          1029  3     TPARSE_BLOCK[TPA$L_STRINGCNT] = .CLI_DESC [DSC$W_LENGTH];
;  615          1030  3     TPARSE_BLOCK[TPA$L_STRINGPTR] = .CLI_DESC [DSC$A_POINTER];
;  616          1031  3     IF NOT LIB$TPARSE (TPARSE_BLOCK, DATACHECK_STB, DATACHECK_KTB)
;  617          1032  3     THEN
;  618          1033  3         ERR_EXIT (INIT$_BADDATCHK);
;  619          1034  3     VALUE_FOUND = 1;
;  620          1035  2 END;
;  621          1036  2
;  622          1037  2 ! If the qualifier /DATA_CHECK was specified with no value, then
;  623          1038  2 ! WRITE data check is the default.  Set the corresponding bit.
;  624          1039  2 !
;  625          1040  2 IF NOT .VALUE_FOUND
;  626          1041  2 THEN
;  627          1042  2     INIT_OPTIONS [OPT_WRITECHECK] = 1;
;  628          1043  2
;  629          1044  2
;  630          1045  1 END;                                    ! end of routine DATACHECK_ACT


                                                        .EXTRN  DATACHECK_STB, DATACHECK_KTB

                                        000C 00000 DATACHECK_ACT:
                                                        .WORD   Save R2,R3                      ; 1009
                        53      0000'   CF  9E 00002            MOVAB   CLI_DESC, R3            ;
                                        52  D4 00007            CLRL    VALUE_FOUND             ; 1025
                                        53  DD 00009 1$:         PUSHL   R3                      ; 1027
                                0000'   CF  9F 0000B            PUSHAB  DATA_DESC               ;
                    0000G       CF          02  FB 0000F        CALLS   #2, CLI$GET_VALUE       ;
                                2E          50  E9 00014        BLBC    R0, 3$                  ;
                      10        A3          63  3C 00017        MOVZWL  CLI_DESC, TPARSE_BLOCK+8        ; 1029
                      14        A3    04    A3  D0 0001B        MOVL    CLI_DESC+4, TPARSE_BLOCK+12     ; 1030
                                0000G   CF  9F 00020            PUSHAB  DATACHECK_KTB           ; 1031
                                0000G   CF  9F 00024            PUSHAB  DATACHECK_STB           ;
                                  08    A3  9F 00028            PUSHAB  TPARSE_BLOCK            ;
                    0000G       CF          03  FB 0002B        CALLS   #3, LIB$TPARSE          ;
                                0D          50  E8 00030        BLBS    R0, 2$                  ;
```

```
                    0075800C  8F  DD 00033          PUSHL   #7700492                          ; 1033
                              01  FB 00039          CALLS   #1, LIB$STOP                    ;
00000000G   00                01  D0 00040 2$:      MOVL    #1, VALUE_FOUND                 ; 1034
            52                C4  11 00043         BRB     1$                              ; 1027
                              52  E8 00045 3$:      BLBS    VALUE_FOUND, 4$                 ; 1040
      0000'  CF               01  88 00048         BISB2   #1, INIT_OPTIONS+1              ; 1042
                                  04 0004D 4$:      RET                                     ; 1045

; Routine Size:  78 bytes,    Routine Base:  $CODE$ + 03D5
```

```
;  632          1046  1 ROUTINE DENSITY_ACT : NOVALUE =
;  633          1047  1
;  634          1048  2 BEGIN
;  635          1049  2
;  636          1050  2 EXTERNAL
;  637          1051  2         DENSITY_STB     : VECTOR [0],   ! state table address
;  638          1052  2         DENSITY_KTB     : VECTOR [0];   ! keyword table address
;  639          1053  2
;  640          1054  2 EXTERNAL ROUTINE
;  641          1055  2         LIB$TPARSE;
;  642          1056  2
;  643          1057  2 ! Parse the DENSITY value and set appropriate flags.
;  644          1058  2 !
;  645          1059  2
;  646          1060  2 CLI$GET_VALUE ( DENSITY_DESC, CLI_DESC );
;  647          1061  2 TPARSE_BLOCK[TPA$L_STRINGCNT] = .CLI_DESC [DSC$W_LENGTH];
;  648          1062  2 TPARSE_BLOCK[TPA$L_STRINGPTR] = .CLI_DESC [DSC$A_POINTER];
;  649          1063  2 IF NOT LIB$TPARSE (TPARSE_BLOCK, DENSITY_STB, DENSITY_KTB)
;  650          1064  2 THEN
;  651          1065  2     ERR_EXIT (INIT$_BADDENS);
;  652          1066  1 END;


                                                .EXTRN  DENSITY_STB, DENSITY_KTB

                                0004 00000 DENSITY_ACT:
                                                .WORD   Save R2                                 ; 1046
                        52      0000'   CF  9E 00002    MOVAB   CLI_DESC, R2                    ;
                                52      DD  00007       PUSHL   R2                              ; 1060
                                0000'   CF  9F 00009    PUSHAB  DENSITY_DESC                    ;
                    0000G   CF          02  FB 0000D    CALLS   #2, CLI$GET_VALUE               ;
                      10    A2          62  3C 00012    MOVZWL  CLI_DESC, TPARSE_BLOCK+8        ; 1061
                      14    A2    04    A2  D0 00016    MOVL    CLI_DESC+4, TPARSE_BLOCK+12     ; 1062
                            0000G   CF  9F 0001B    PUSHAB  DENSITY_KTB                     ; 1063
                            0000G   CF  9F 0001F    PUSHAB  DENSITY_STB                     ;
                              08    A2  9F 00023    PUSHAB  TPARSE_BLOCK                    ;
                    0000G   CF          03  FB 00026    CALLS   #3, LIB$TPARSE                  ;
                            0D          50  E8 0002B    BLBS    R0, 1$                          ;
                                00758014    8F  DD 0002E    PUSHL   #7700500                        ; 1065
                00000000G   00          01  FB 00034    CALLS   #1, LIB$STOP                    ;
                                            04  0003B 1$:   RET                                 ; 1066

; Routine Size:  60 bytes,    Routine Base:  $CODE$ + 0423

;  653          1067  1
```

```
;  655          1068  1 ROUTINE FILE_PROT_ACT (REQ_DESC, CLI_CALLBACK) : NOVALUE =
;  656          1069  2 BEGIN
;  657          1070  2
;  658          1071  2 EXTERNAL
;  659          1072  2         PROTECTION_STB  : VECTOR [0],   ! state table address
;  660          1073  2         PROTECTION_KTB  : VECTOR [0];   ! keyword table address
;  661          1074  2
;  662          1075  2 EXTERNAL ROUTINE
;  663          1076  2         LIB$TPARSE;
;  664          1077  2
;  665          1078  2 ! Parse the switch value string (storing the binary protection).
;  666          1079  2 ! Complement thereafter, since the parser produces the complement.
;  667          1080  2 !
;  668          1081  2
;  669          1082  2 PROT_VAL = 0;
;  670          1083  2
;  671          1084  2 WHILE CLI$GET_VALUE ( FILE_DESC, CLI_DESC ) DO
;  672          1085  3 BEGIN
;  673          1086  3     TPARSE_BLOCK[TPA$L_STRINGCNT] = .CLI_DESC [DSC$W_LENGTH];
;  674          1087  3     TPARSE_BLOCK[TPA$L_STRINGPTR] = .CLI_DESC [DSC$A_POINTER];
;  675          1088  3     IF NOT LIB$TPARSE (TPARSE_BLOCK, PROTECTION_STB, PROTECTION_KTB)
;  676          1089  3     THEN
;  677          1090  3         ERR_EXIT (INIT$_BADPRO);
;  678          1091  2 END;
;  679          1092  2
;  680          1093  2 FILE_PROT = NOT .PROT_VAL;
;  681          1094  2
;  682          1095  1 END;                                    ! end of routine FILE_PROT_ACT


                                                        .EXTRN  PROTECTION_STB, PROTECTION_KTB

                                        0004 00000 FILE_PROT_ACT:
                                                        .WORD   Save R2                                 ; 1068
                                52      0000'  CF  9E 00002     MOVAB   PROT_VAL, R2                    ;
                                               62  D4 00007     CLRL    PROT_VAL                        ; 1082
                                          D4   A2  9F 00009 1$: PUSHAB  CLI_DESC                        ; 1084
                                        0000'  CF  9F 0000C     PUSHAB  FILE_DESC                       ;
                          0000G CF                 02  FB 00010     CALLS   #2, CLI$GET_VALUE               ;
                                2C                 50  E9 00015     BLBC    R0, 2$                  ;
                             E4 A2          D4     A2  3C 00018     MOVZWL  CLI_DESC, TPARSE_BLOCK+8        ; 1086
                             E8 A2          D8     A2  D0 0001D     MOVL    CLI_DESC+4, TPARSE_BLOCK+12     ; 1087
                                        0000G  CF  9F 00022     PUSHAB  PROTECTION_KTB                  ; 1088
                                        0000G  CF  9F 00026     PUSHAB  PROTECTION_STB                  ;
                                          DC   A2  9F 0002A     PUSHAB  TPARSE_BLOCK                    ;
                          0000G CF                 03  FB 0002D     CALLS   #3, LIB$TPARSE                  ;
                                D4                 50  E8 00032     BLBS    R0, 1$                  ;
                                     0075801C      8F  DD 00035     PUSHL   #7700508                        ; 1090
                    00000000G   00               01  FB 0003B     CALLS   #1, LIB$STOP                    ;
                                               C5  11 00042     BRB     1$                              ; 1084
                          0000' CF               62  D2 00044 2$: MCOML   PROT_VAL, FILE_PROT             ; 1093
                                                 04  00049     RET                                     ; 1095

; Routine Size:  74 bytes,    Routine Base:  $CODE$ + 045F
```

```
;  684          1096  1 ROUTINE INDEX_ACT (REQ_DESC, CLI_CALLBACK) : NOVALUE =
;  685          1097  2 BEGIN
;  686          1098  2
;  687          1099  2 EXTERNAL
;  688          1100  2         INDEX_STB       : VECTOR [0],   ! state table address
;  689          1101  2         INDEX_KTB       : VECTOR [0];   ! keyword table address
;  690          1102  2
;  691          1103  2 EXTERNAL ROUTINE
;  692          1104  2         LIB$TPARSE;
;  693          1105  2
;  694          1106  2 ! Parse the INDEX options string.
;  695          1107  2 !
;  696          1108  2
;  697          1109  2 CLI$GET_VALUE ( INDEX_DESC, CLI_DESC );
;  698          1110  2 TPARSE_BLOCK[TPA$L_STRINGCNT] = .CLI_DESC [DSC$W_LENGTH];
;  699          1111  2 TPARSE_BLOCK[TPA$L_STRINGPTR] = .CLI_DESC [DSC$A_POINTER];
;  700          1112  2 IF NOT LIB$TPARSE (TPARSE_BLOCK, INDEX_STB, INDEX_KTB)
;  701          1113  2 THEN ERR_EXIT (INIT$_BADINDEX);
;  702          1114  2
;  703          1115  1 END;                                     ! end of routine INDEX_ACT


                                                        .EXTRN  INDEX_STB, INDEX_KTB

                                    0004 00000 INDEX_ACT:
                                                        .WORD   Save R2                                 ; 1096
                          52    0000'  CF  9E 00002             MOVAB   CLI_DESC, R2                    ;
                                       52  DD 00007             PUSHL   R2                              ; 1109
                                0000'  CF  9F 00009             PUSHAB  INDEX_DESC                      ;
                      0000G     CF         02  FB 0000D         CALLS   #2, CLI$GET_VALUE               ;
                        10      A2         62  3C 00012         MOVZWL  CLI_DESC, TPARSE_BLOCK+8        ; 1110
                        14      A2   04    A2  D0 00016         MOVL    CLI_DESC+4, TPARSE_BLOCK+12     ; 1111
                                0000G  CF  9F  0001B            PUSHAB  INDEX_KTB                       ; 1112
                                0000G  CF  9F  0001F            PUSHAB  INDEX_STB                       ;
                                   08  A2  9F  00023            PUSHAB  TPARSE_BLOCK                    ;
                      0000G     CF         03  FB 00026         CALLS   #3, LIB$TPARSE                  ;
                                   0D         50  E8 0002B         BLBS    R0, 1$                      ;
                                0075808C   8F  DD 0002E            PUSHL   #7700620                    ; 1113
                00000000G       00         01  FB 00034         CALLS   #1, LIB$STOP                    ;
                                           04  0003B 1$:        RET                                     ; 1115

; Routine Size:  60 bytes,    Routine Base:  $CODE$ + 04A9
```

```
;  705          1116  1 ROUTINE LABEL_QUAL_ACT (REQ_DESC, CLI_CALLBACK) : NOVALUE =
;  706          1117  2 BEGIN
;  707          1118  2
;  708          1119  2 EXTERNAL
;  709          1120  2         LABEL_QUAL_STB  : VECTOR [0],   ! state table address
;  710          1121  2         LABEL_QUAL_KTB  : VECTOR [0];   ! keyword table address
;  711          1122  2
;  712          1123  2 EXTERNAL ROUTINE
;  713          1124  2         LIB$TPARSE;
;  714          1125  2
;  715          1126  2 DATA_INDEX = 0;                  ! set index to zero.
;  716          1127  2
;  717          1128  2 ! Parse the LABEL options string.
;  718          1129  2 !
;  719          1130  2
;  720          1131  2 WHILE CLI$GET_VALUE ( LABEL_DESC, CLI_DESC ) DO
;  721          1132  3 BEGIN
;  722          1133  3     TPARSE_BLOCK[TPA$L_STRINGCNT] = .CLI_DESC [DSC$W_LENGTH];
;  723          1134  3     TPARSE_BLOCK[TPA$L_STRINGPTR] = .CLI_DESC [DSC$A_POINTER];
;  724          1135  3     IF NOT LIB$TPARSE (TPARSE_BLOCK, LABEL_QUAL_STB, LABEL_QUAL_KTB)
;  725          1136  3     THEN
;  726          1137  3         ERR_EXIT (INIT$_BADLABELQ);
;  727          1138  2 END;
;  728          1139  2
;  729          1140  1 END;                                    ! end of routine LABEL_QUAL_ACT


                                                .EXTRN  LABEL_QUAL_STB, LABEL_QUAL_KTB

                                0004 00000 LABEL_QUAL_ACT:
                                           .WORD   Save R2                                  ;  1116
                        52      0000'  CF  9E 00002        MOVAB   CLI_DESC, R2             ;
                               0000'   CF  D4 00007        CLRL    DATA_INDEX               ;  1126
                                       52  DD 0000B 1$:     PUSHL   R2                       ;  1131
                               0000'   CF  9F 0000D        PUSHAB  LABEL_DESC               ;
                0000G   CF             02  FB 00011        CALLS   #2, CLI$GET_VALUE        ;
                        2B             50  E9 00016        BLBC    R0, 2$                   ;
                  10    A2             62  3C 00019        MOVZWL  CLI_DESC, TPARSE_BLOCK+8 ;  1133
                  14    A2       04    A2  D0 0001D        MOVL    CLI_DESC+4, TPARSE_BLOCK+12 ;  1134
                               0000G   CF  9F 00022        PUSHAB  LABEL_QUAL_KTB           ;  1135
                               0000G   CF  9F 00026        PUSHAB  LABEL_QUAL_STB           ;
                                 08    A2  9F 0002A        PUSHAB  TPARSE_BLOCK             ;
                0000G   CF             03  FB 0002D        CALLS   #3, LIB$TPARSE           ;
                        D6             50  E8 00032        BLBS    R0, 1$                   ;
                           00758054    8F  DD 00035        PUSHL   #7700564                 ;  1137
            00000000G   00             01  FB 0003B        CALLS   #1, LIB$STOP             ;
                                       C7  11 00042        BRB     1$                       ;  1131
                                           04 00044 2$:    RET                              ;  1140

; Routine Size:  69 bytes,    Routine Base:  $CODE$ + 04E5
```

```
;   731          1141  1 ROUTINE OVERRIDE_ACT : NOVALUE =
;   732          1142  2 BEGIN
;   733          1143  2
;   734          1144  2 EXTERNAL
;   735          1145  2         OVERRIDE_STB    : VECTOR [0],   ! state table address
;   736          1146  2         OVERRIDE_KTB    : VECTOR [0];   ! keyword table address
;   737          1147  2
;   738          1148  2 EXTERNAL ROUTINE
;   739          1149  2         LIB$TPARSE;
;   740          1150  2
;   741          1151  2 ! Parse the OVERRIDE string and set appropriate flags.
;   742          1152  2 !
;   743          1153  2
;   744          1154  2 WHILE CLI$GET_VALUE ( OVERRIDE_DESC, CLI_DESC ) DO
;   745          1155  3 BEGIN
;   746          1156  3     TPARSE_BLOCK[TPA$L_STRINGCNT] = .CLI_DESC [DSC$W_LENGTH];
;   747          1157  3     TPARSE_BLOCK[TPA$L_STRINGPTR] = .CLI_DESC [DSC$A_POINTER];
;   748          1158  3     IF NOT LIB$TPARSE (TPARSE_BLOCK, OVERRIDE_STB, OVERRIDE_KTB)
;   749          1159  3     THEN
;   750          1160  3         ERR_EXIT (INIT$_BADOVR);
;   751          1161  2 END;
;   752          1162  2
;   753          1163  1 END;                                    ! end of routine OVERRIDE_ACT


                                                        .EXTRN  OVERRIDE_STB, OVERRIDE_KTB

                                        0004 00000 OVERRIDE_ACT:
                                                   .WORD   Save R2                                  ; 1141
                        52      0000'   CF  9E 00002       MOVAB   CLI_DESC, R2                     ;
                                        52  DD 00007 1$:     PUSHL   R2                           ; 1154
                                0000'   CF  9F 00009       PUSHAB  OVERRIDE_DESC                    ;
              0000G     CF              02  FB 0000D       CALLS   #2, CLI$GET_VALUE                ;
                        2B              50  E9 00012       BLBC    R0, 2$                           ;
                10      A2              62  3C 00015       MOVZWL  CLI_DESC, TPARSE_BLOCK+8         ; 1156
                14      A2          04  A2  D0 00019       MOVL    CLI_DESC+4, TPARSE_BLOCK+12      ; 1157
                            0000G   CF  9F 0001E       PUSHAB  OVERRIDE_KTB                     ; 1158
                            0000G   CF  9F 00022       PUSHAB  OVERRIDE_STB                     ;
                              08    A2  9F 00026       PUSHAB  TPARSE_BLOCK                     ;
              0000G     CF              03  FB 00029       CALLS   #3, LIB$TPARSE                   ;
                        D6              50  E8 0002E       BLBS    R0, 1$                           ;
                            007580CC    8F  DD 00031       PUSHL   #7700684                         ; 1160
          00000000G     00              01  FB 00037       CALLS   #1, LIB$STOP                     ;
                                        C7  11 0003E       BRB     1$                               ; 1154
                                            04 00040 2$:   RET                                      ; 1163

; Routine Size:  65 bytes,    Routine Base:  $CODE$ + 052A
```

```
;  755          1164  1 ROUTINE OWNER_UIC_ACT : NOVALUE =
;  756          1165  2 BEGIN
;  757          1166  2
;  758          1167  2 EXTERNAL
;  759          1168  2         UIC_STB         : VECTOR [0],   ! state table address
;  760          1169  2         UIC_KTB         : VECTOR [0];   ! keyword table address
;  761          1170  2
;  762          1171  2 EXTERNAL ROUTINE
;  763          1172  2         LIB$TPARSE;
;  764          1173  2
;  765          1174  2 ! Parse the UIC string and store it in the owner UIC longword.
;  766          1175  2 !
;  767          1176  2
;  768          1177  2 WHILE CLI$GET_VALUE ( OWNER_DESC, CLI_DESC ) DO
;  769          1178  3 BEGIN
;  770          1179  3     TPARSE_BLOCK[TPA$L_STRINGCNT] = .CLI_DESC [DSC$W_LENGTH];
;  771          1180  3     TPARSE_BLOCK[TPA$L_STRINGPTR] = .CLI_DESC [DSC$A_POINTER];
;  772          1181  3     IF NOT LIB$TPARSE (TPARSE_BLOCK, UIC_STB, UIC_KTB)
;  773          1182  3     THEN
;  774          1183  3         ERR_EXIT (INIT$_BADUIC);
;  775          1184  2 END;
;  776          1185  2
;  777          1186  1 END;                                    ! end of routine OWNER_UIC_ACT


                                                        .EXTRN  UIC_STB, UIC_KTB

                                0004 00000 OWNER_UIC_ACT:
                                                        .WORD   Save R2                                 ; 1164
                          52    0000'  CF  9E 00002             MOVAB   CLI_DESC, R2                    ;
                                       52  DD 00007 1$:          PUSHL   R2                              ; 1177
                                0000'  CF  9F 00009             PUSHAB  OWNER_DESC                      ;
                     0000G      CF         02  FB 0000D         CALLS   #2, CLI$GET_VALUE               ;
                                2B         50  E9 00012         BLBC    R0, 2$                          ;
                       10       A2         62  3C 00015         MOVZWL  CLI_DESC, TPARSE_BLOCK+8        ; 1179
                       14       A2   04    A2  D0 00019         MOVL    CLI_DESC+4, TPARSE_BLOCK+12     ; 1180
                                0000G  CF  9F 0001E             PUSHAB  UIC_KTB                         ; 1181
                                0000G  CF  9F 00022             PUSHAB  UIC_STB                         ;
                                  08   A2  9F 00026             PUSHAB  TPARSE_BLOCK                    ;
                     0000G      CF         03  FB 00029         CALLS   #3, LIB$TPARSE                  ;
                                D6         50  E8 0002E         BLBS    R0, 1$                          ;
                                00758024   8F  DD 00031         PUSHL   #7700516                        ; 1183
                00000000G       00        01  FB 00037         CALLS   #1, LIB$STOP                    ;
                                          C7  11 0003E         BRB     1$                              ; 1177
                                              04 00040 2$:     RET                                     ; 1186

; Routine Size:  65 bytes,    Routine Base:  $CODE$ + 056B
```

```
;  779          1187  1 ROUTINE PROTECTION_ACT : NOVALUE =
;  780          1188  1
;  781          1189  2 BEGIN
;  782          1190  2
;  783          1191  2 EXTERNAL
;  784          1192  2         PROTECTION_STB  : VECTOR [0],   ! state table address
;  785          1193  2         PROTECTION_KTB  : VECTOR [0];   ! keyword table address
;  786          1194  2
;  787          1195  2 EXTERNAL ROUTINE
;  788          1196  2         LIB$TPARSE;
;  789          1197  2
;  790          1198  2 ! Parse the PROTECTION qualifier string storing the binary protection.
;  791          1199  2 ! Complement thereafter, since the parser produces the complement.
;  792          1200  2 !
;  793          1201  2
;  794          1202  2 PROT_VAL = 0;
;  795          1203  2
;  796          1204  2 WHILE CLI$GET_VALUE ( PROTECTION_DESC, CLI_DESC ) DO
;  797          1205  3 BEGIN
;  798          1206  3     TPARSE_BLOCK[TPA$L_STRINGCNT] = .CLI_DESC [DSC$W_LENGTH];
;  799          1207  3     TPARSE_BLOCK[TPA$L_STRINGPTR] = .CLI_DESC [DSC$A_POINTER];
;  800          1208  3     IF NOT LIB$TPARSE (TPARSE_BLOCK, PROTECTION_STB, PROTECTION_KTB)
;  801          1209  3     THEN
;  802          1210  3         ERR_EXIT (INIT$_BADPRO);
;  803          1211  2 END;
;  804          1212  2
;  805          1213  2 PROTECTION = NOT .PROT_VAL;
;  806          1214  2
;  807          1215  1 END;


                                0004 00000 PROTECTION_ACT:
                                           .WORD   Save R2                                          ; 1187
                     52   0000'  CF   9E 00002          MOVAB   PROT_VAL, R2                      ;
                                 62   D4 00007          CLRL    PROT_VAL                          ; 1202
                                   D4 A2 9F 00009 1$:    PUSHAB  CLI_DESC                          ; 1204
                              0000' CF   9F 0000C        PUSHAB  PROTECTION_DESC                   ;
                0000G CF           02   FB 00010        CALLS   #2, CLI$GET_VALUE                 ;
                      2C           50   E9 00015        BLBC    R0, 2$                            ;
                   E4 A2     D4    A2   3C 00018        MOVZWL  CLI_DESC, TPARSE_BLOCK+8          ; 1206
                   E8 A2     D8    A2   D0 0001D        MOVL    CLI_DESC+4, TPARSE_BLOCK+12       ; 1207
                              0000G CF   9F 00022        PUSHAB  PROTECTION_KTB                    ; 1208
                              0000G CF   9F 00026        PUSHAB  PROTECTION_STB                    ;
                                 DC A2   9F 0002A        PUSHAB  TPARSE_BLOCK                      ;
                0000G CF           03   FB 0002D        CALLS   #3, LIB$TPARSE                    ;
                      D4           50   E8 00032        BLBS    R0, 1$                            ;
                           0075801C 8F   DD 00035        PUSHL   #7700508                          ; 1210
           00000000G   00           01   FB 0003B        CALLS   #1, LIB$STOP                      ;
                                 C5   11 00042        BRB     1$                                ; 1204
                0000' CF           62   D2 00044 2$:     MCOML   PROT_VAL, PROTECTION              ; 1213
                                      04 00049          RET                                       ; 1215

; Routine Size:  74 bytes,    Routine Base:  $CODE$ + 05AC
```

```
;  809          1216  1 !
;  810          1217  1 ! Specify disk structure level
;  811          1218  1 !
;  812          1219  1 ROUTINE STRUCTURE_ACT (REQ_DESC, CLI_CALLBACK) : NOVALUE =
;  813          1220  2 BEGIN
;  814          1221  2
;  815          1222  2 EXTERNAL
;  816          1223  2         STRUCTURE_STB   : VECTOR [0],   ! state table address
;  817          1224  2         STRUCTURE_KTB   : VECTOR [0];   ! keyword table address
;  818          1225  2
;  819          1226  2 EXTERNAL ROUTINE
;  820          1227  2         LIB$TPARSE;
;  821          1228  2
;  822          1229  2 ! Parse the STRUCTURE string (setting the level 1 bit if so).
;  823          1230  2 !
;  824          1231  2
;  825          1232  2 CLI$GET_VALUE ( STRUCTURE_DESC, CLI_DESC );
;  826          1233  2 TPARSE_BLOCK[TPA$L_STRINGCNT] = .CLI_DESC [DSC$W_LENGTH];
;  827          1234  2 TPARSE_BLOCK[TPA$L_STRINGPTR] = .CLI_DESC [DSC$A_POINTER];
;  828          1235  2 IF NOT LIB$TPARSE (TPARSE_BLOCK, STRUCTURE_STB, STRUCTURE_KTB)
;  829          1236  2 THEN
;  830          1237  2     ERR_EXIT (INIT$_BADSTRUCT);
;  831          1238  2
;  832          1239  1 END;                                    ! end of routine STRUCTURE_ACT


                                                        .EXTRN  STRUCTURE_STB, STRUCTURE_KTB

                                        0004 00000 STRUCTURE_ACT:
                                                        .WORD   Save R2                                 ; 1219
                             52   0000'  CF   9E 00002         MOVAB   CLI_DESC, R2                    ;
                                         52   DD 00007         PUSHL   R2                              ; 1232
                                   0000'  CF   9F 00009         PUSHAB  STRUCTURE_DESC                  ;
                        0000G    CF             02   FB 0000D  CALLS   #2, CLI$GET_VALUE               ;
                           10    A2             62   3C 00012  MOVZWL  CLI_DESC, TPARSE_BLOCK+8        ; 1233
                           14    A2       04    A2   D0 00016  MOVL    CLI_DESC+4, TPARSE_BLOCK+12     ; 1234
                                   0000G  CF   9F 0001B         PUSHAB  STRUCTURE_KTB                   ; 1235
                                   0000G  CF   9F 0001F         PUSHAB  STRUCTURE_STB                   ;
                                     08   A2   9F 00023         PUSHAB  TPARSE_BLOCK                    ;
                        0000G    CF             03   FB 00026  CALLS   #3, LIB$TPARSE                  ;
                                 0D             50   E8 0002B  BLBS    R0, 1$                          ;
                                      007580E4   8F   DD 0002E  PUSHL   #7700708                       ; 1237
                    00000000G    00             01   FB 00034  CALLS   #1, LIB$STOP                   ;
                                                04 0003B 1$:    RET                                     ; 1239

; Routine Size:  60 bytes,    Routine Base:  $CODE$ + 05F6
```

```
;  834          1240  1 !
;  835          1241  1 ! Get volume user name
;  836          1242  1 !
;  837          1243  1 ROUTINE USER_NAME_ACT (REQ_DESC, CLI_CALLBACK) : NOVALUE =
;  838          1244  2 BEGIN
;  839          1245  2
;  840          1246  2 ! Store the descriptor for the volume USER_NAME.
;  841          1247  2 !
;  842          1248  2
;  843          1249  2 CH$FILL ( 0, DSC$C_S_BLN, USER_NAME );
;  844          1250  2 USER_NAME[DSC$B_CLASS] = DSC$K_CLASS_D;
;  845          1251  2 CLI$GET_VALUE (USER_DESC, USER_NAME);
;  846          1252  2
;  847          1253  1 END;                                    ! end of routine USER_NAME_ACT


                                        003C 00000 USER_NAME_ACT:
                                                   .WORD   Save R2,R3,R4,R5                        ; 1243
        08              00        6E              00  2C 00002  MOVC5   #0, (SP), #0, #8, USER_NAME  ; 1249
                                            0000'   CF         00007
                                  0000'   CF       02  90 0000A  MOVB    #2, USER_NAME+3           ; 1250
                                            0000'   CF  9F 0000F  PUSHAB  USER_NAME                 ; 1251
                                            0000'   CF  9F 00013  PUSHAB  USER_DESC
                                  0000G   CF       02  FB 00017  CALLS   #2, CLI$GET_VALUE
                                                   04  0001C  RET                                 ; 1253

; Routine Size:  29 bytes,    Routine Base:  $CODE$ + 0632
```

```
;  849          1254  1 !
;  850          1255  1 ! Store index file LBN
;  851          1256  1 !
;  852          1257  1 ROUTINE GET_INDEX_LBN =
;  853          1258  2 BEGIN
;  854          1259  2
;  855          1260  2 TPARSE_ARGS (CONTEXT);
;  856          1261  2
;  857          1262  2 INDEX = .CONTEXT[TPA$L_NUMBER];
;  858          1263  2 RETURN 1;
;  859          1264  2
;  860          1265  1 END;                                    ! end of routine GET_INDEX_LBN


                                        0000 00000 GET_INDEX_LBN:
                                                   .WORD   Save nothing                          ; 1257
                    0000'  CF       1C  AC  D0 00002       MOVL    28(CONTEXT), INDEX           ; 1262
                           50           01  D0 00008       MOVL    #1, R0                       ; 1263
                                            04 0000B       RET                                  ; 1265

; Routine Size:  12 bytes,    Routine Base:  $CODE$ + 064F
```

```
;  862          1266  1 !
;  863          1267  1 ! Store bad block LBN or sector number.
;  864          1268  1 !
;  865          1269  1 ROUTINE GET_BAD_LBN =
;  866          1270  2 BEGIN
;  867          1271  2
;  868          1272  2 TPARSE_ARGS (CONTEXT);
;  869          1273  2
;  870          1274  2 IF .BADBLOCK_COUNT GEQ BAD_TABLE_LEN
;  871          1275  2 THEN ERR_EXIT (INIT$_MAXBAD);
;  872          1276  2 BADBLOCK_TABLE[.BADBLOCK_COUNT, BAD_LBN] = .CONTEXT[TPA$L_NUMBER];
;  873          1277  2 BADBLOCK_TABLE[.BADBLOCK_COUNT, BAD_COUNT] = 1;
;  874          1278  2 BADBLOCK_COUNT = .BADBLOCK_COUNT + 1;
;  875          1279  2 RETURN 1;
;  876          1280  2
;  877          1281  1 END;                                    ! end of routine GET_BAD_LBN


                                0004 00000 GET_BAD_LBN:
                                           .WORD   Save R2                                          ; 1269
                     52   0000'  CF  9E 00002          MOVAB   BADBLOCK_COUNT, R2
          00000064   8F           62  D1 00007          CMPL    BADBLOCK_COUNT, #100                ; 1274
                                  0D  19 0000E          BLSS    1$
                     007580BC     8F  DD 00010          PUSHL   #7700668                            ; 1275
          00000000G  00           01  FB 00016          CALLS   #1, LIB$STOP
                     50           62  D0 0001D 1$:      MOVL    BADBLOCK_COUNT, R0                  ; 1276
                            04 A240  7F 00020          PUSHAQ  BADBLOCK_TABLE[R0]
                     9E        1C   AC  D0 00024          MOVL    28(CONTEXT), @(SP)+
                            08 A240  7F 00028          PUSHAQ  BADBLOCK_TABLE+4[R0]                ; 1277
                     9E           01  B0 0002C          MOVW    #1, @(SP)+
                                  62  D6 0002F          INCL    BADBLOCK_COUNT                      ; 1278
                     50           01  D0 00031          MOVL    #1, R0                              ; 1279
                                      04 00034          RET                                         ; 1281

; Routine Size:  53 bytes,    Routine Base:  $CODE$ + 065B
```

```
;  879          1282  1 !
;  880          1283  1 ! Store bad block track number.
;  881          1284  1 !
;  882          1285  1 ROUTINE GET_BAD_TRACK =
;  883          1286  2 BEGIN
;  884          1287  2
;  885          1288  2 TPARSE_ARGS (CONTEXT);
;  886          1289  2
;  887          1290  2 BADBLOCK_TABLE[.BADBLOCK_COUNT-1, BAD_TRACK] = .CONTEXT[TPA$L_NUMBER];
;  888          1291  2 BADBLOCK_TABLE[.BADBLOCK_COUNT-1, BAD_STC_FORM] = 1;
;  889          1292  2 RETURN 1;
;  890          1293  2
;  891          1294  1 END;                                    ! end of routine GET_BAD_TRACK


                                        0000 00000 GET_BAD_TRACK:
                                                   .WORD   Save nothing                          ; 1285
                              50   0000'   CF  D0 00002     MOVL    BADBLOCK_COUNT, R0         ; 1290
                                   0000'CF40  7F 00007     PUSHAQ  BADBLOCK_TABLE-7[R0]
                              9E     1C    AC  90 0000C     MOVB    28(CONTEXT), @(SP)+
                                   0000'CF40  7F 00010     PUSHAQ  BADBLOCK_TABLE-2[R0]       ; 1291
                              9E         01  88 00015     BISB2   #1, @(SP)+
                              50         01  D0 00018     MOVL    #1, R0                     ; 1292
                                             04 0001B     RET                                ; 1294

; Routine Size:  28 bytes,    Routine Base:  $CODE$ + 0690
```

```
;  893          1295  1 !
;  894          1296  1 ! Store bad block cylinder number.
;  895          1297  1 !
;  896          1298  1 ROUTINE GET_BAD_CYL =
;  897          1299  2 BEGIN
;  898          1300  2
;  899          1301  2 TPARSE_ARGS (CONTEXT);
;  900          1302  2
;  901          1303  2 BADBLOCK_TABLE[.BADBLOCK_COUNT-1, BAD_CYLINDER] = .CONTEXT[TPA$L_NUMBER];
;  902          1304  2 RETURN 1;
;  903          1305  2
;  904          1306  1 END;                                    ! end of routine GET_BAD_CYL


                                0000 00000 GET_BAD_CYL:
                                           .WORD   Save nothing                                    ; 1298
                     50   0000'  CF  D0 00002    MOVL    BADBLOCK_COUNT, R0                      ; 1303
                          0000'CF40  7F 00007    PUSHAQ  BADBLOCK_TABLE-6[R0]                    ;
                     9E     1C   AC  B0 0000C    MOVW    28(CONTEXT), @(SP)+                     ;
                     50          01  D0 00010    MOVL    #1, R0                                  ; 1304
                                     04 00013    RET                                             ; 1306

; Routine Size:  20 bytes,    Routine Base:  $CODE$ + 06AC
```

```
;  906          1307  1 !
;  907          1308  1 ! Store bad block count.
;  908          1309  1 !
;  909          1310  1 ROUTINE GET_BAD_COUNT =
;  910          1311  2 BEGIN
;  911          1312  2
;  912          1313  2 TPARSE_ARGS (CONTEXT);
;  913          1314  2
;  914          1315  2 BADBLOCK_TABLE[.BADBLOCK_COUNT-1, BAD_COUNT] = .CONTEXT[TPA$L_NUMBER];
;  915          1316  2 IF .CONTEXT[TPA$L_NUMBER] EQL 0
;  916          1317  2 THEN ERR_EXIT (INIT$_BADBLOCKS);
;  917          1318  2 RETURN 1;
;  918          1319  2
;  919          1320  1 END;                                    ! end of routine GET_BAD_COUNT


                                    0000 00000 GET_BAD_COUNT:
                                               .WORD   Save nothing                                    ;  1310
                          50     0000'   CF  D0 00002          MOVL    BADBLOCK_COUNT, R0              ;  1315
                                 0000'CF40  7F 00007          PUSHAQ  BADBLOCK_TABLE-4[R0]            ;
                          9E       1C    AC  B0 0000C          MOVW    28(CONTEXT), @(SP)+             ;
                                   1C    AC  D5 00010          TSTL    28(CONTEXT)                     ;  1316
                                         0D  12 00013          BNEQ    1$                              ;
                             00758084    8F  DD 00015          PUSHL   #7700612                        ;  1317
                 00000000G   00          01  FB 0001B          CALLS   #1, LIB$STOP                    ;
                          50             01  D0 00022 1$:      MOVL    #1, R0                          ;  1318
                                            04 00025          RET                                     ;  1320

; Routine Size:  38 bytes,    Routine Base:  $CODE$ + 06C0
```

```
;  921          1321  1 !
;  922          1322  1 ! Determine if ANSI VOL1 accessibility character is an ANSI 'a' character
;  923          1323  1 !
;  924          1324  1 ROUTINE GET_VOL_ACC =
;  925          1325  2 BEGIN
;  926          1326  2
;  927          1327  2 TPARSE_ARGS (CONTEXT);
;  928          1328  2
;  929          1329  2 VOL_ACC = .(.CONTEXT[TPA$L_TOKENPTR]);
;  930          1330  2 SELECTONE .VOL_ACC OF
;  931          1331  2     SET
;  932          1332  2     ['A' TO 'Z'] ;;
;  933          1333  2     [' ' TO '"'] ;;
;  934          1334  2     ['%' TO '?'] ;;
;  935          1335  2     ['_']::
;  936          1336  2     ['a' TO 'z'] : VOL_ACC = .VOL_ACC - ( 'a' - 'A' );  . uppercase
;  937          1337  2     [OTHERWISE]  : ERR_EXIT (INIT$_BADVOLACC);
;  938          1338  2     TES;
;  939          1339  2 RETURN 1;
;  940          1340  1 END;


                                        0004 00000 GET_VOL_ACC:
                                                   .WORD   Save R2                                  ; 1324
                                52      0000'  CF  9E 00002         MOVAB   VOL_ACC, R2                      ;
                                62        14   BC  90 00007         MOVB    @20(CONTEXT), VOL_ACC            ; 1329
                                50             62  9A 0000B         MOVZBL  VOL_ACC, R0                      ; 1330
                           41   8F             50  91 0000E         CMPB    R0, #65                          ; 1332
                                               06  1F 00012         BLSSU   1$                               ;
                           5A   8F             50  91 00014         CMPB    R0, #90                          ;
                                               38  1B 00018         BLEQU   5$                               ;
                                20             50  91 0001A 1$:     CMPB    R0, #32                          ; 1333
                                               05  1F 0001D         BLSSU   2$                               ;
                                22             50  91 0001F         CMPB    R0, #34                          ;
                                               2E  1B 00022         BLEQU   5$                               ;
                                25             50  91 00024 2$:     CMPB    R0, #37                          ; 1334
                                               05  1F 00027         BLSSU   3$                               ;
                                3F             50  91 00029         CMPB    R0, #63                          ;
                                               24  1B 0002C         BLEQU   5$                               ;
                           5F   8F             50  91 0002E 3$:     CMPB    R0, #95                          ; 1335
                                               1E  13 00032         BEQL    5$                               ;
                           61   8F             50  91 00034         CMPB    R0, #97                          ; 1336
                                               0B  1F 00038         BLSSU   4$                               ;
                           7A   8F             50  91 0003A         CMPB    R0, #122                         ;
                                               05  1A 0003E         BGTRU   4$                               ;
                                62             20  82 00040         SUBB2   #32, VOL_ACC                     ;
                                               0D  11 00043         BRB     5$                               ;
                                   007580F4    8F  DD 00045 4$:     PUSHL   #7700724                         ; 1337
                  00000000G    00             01  FB 0004B         CALLS   #1, LIB$STOP                     ;
                                50             01  D0 00052 5$:     MOVL    #1, R0                           ; 1339
                                               04  00055         RET                                      ; 1340

; Routine Size:  86 bytes,    Routine Base:  $CODE$ + 06E6
```

```
;  942          1341  1 !
;  943          1342  1 ! Determine if ANSI VOL1 OWNER IDENTIFIER is ANSI "a" characters
;  944          1343  1 !
;  945          1344  1 ROUTINE FORMAT_VOL_OWNER =
;  946          1345  2 BEGIN
;  947          1346  2
;  948          1347  2 LOCAL
;  949          1348  2         TEMP : BYTE,
;  950          1349  2         J,
;  951          1350  2         I;
;  952          1351  2
;  953          1352  2 TPARSE_ARGS (CONTEXT);
;  954          1353  2
;  955          1354  2 CH$FILL (' ',14,VOL_OWNER);
;  956          1355  2 J = 0;
;  957          1356  2 I = 0;
;  958          1357  2 TEMP = .(.CONTEXT[TPA$L_TOKENPTR]);
;  959          1358  3 IF NOT (.TEMP EQL ',')
;  960          1359  2    THEN
;  961          1360  3    BEGIN
;  962          1361  3         CH$MOVE (.CONTEXT[TPA$L_TOKENCNT], .CONTEXT[TPA$L_TOKENPTR],
;  963          1362  3             DATA_PTR[.DATA_INDEX]);
;  964          1363  3         DATA_INDEX = .DATA_INDEX + .CONTEXT[TPA$L_TOKENCNT];
;  965          1364  2    END;
;  966          1365  2 WHILE .I LEQ (.DATA_INDEX - 1) DO
;  967          1366  3    BEGIN
;  968          1367  3         LOCAL CHAR : BYTE;
;  969          1368  3         CHAR = .DATA_PTR[.I];
;  970          1369  3         SELECTONE .CHAR OF
;  971          1370  3             SET
;  972          1371  3             ['A' TO 'Z'] :;
;  973          1372  3             [' ' TO '"'] :;
;  974          1373  3             ['%' TO '?'] :;
;  975          1374  3             ['_']:;
;  976          1375  3             ['a' TO 'z'] : CHAR = .CHAR - ( 'a' - 'A' ); ! uppercase
;  977          1376  3             [OTHERWISE]  : ERR_EXIT (INIT$_BADOWNID);
;  978          1377  3             TES;
;  979          1378  4         IF (.CHAR EQL '"') AND (.I EQL (.DATA_INDEX - 1))
;  980          1379  3            THEN RETURN 1;
;  981          1380  4         IF NOT ((.CHAR EQL '"') AND (.I EQL 0))
;  982          1381  3            THEN
;  983          1382  4            BEGIN
;  984          1383  4                 VOL_OWNER[.J] = .CHAR;
;  985          1384  4                 J = .J + 1;
;  986          1385  5                 IF NOT (.CHAR EQL '"' AND .DATA_PTR[.I+1] EQL '"')
;  987          1386  4                    THEN I = .I + 1
;  988          1387  4                    ELSE I = .I + 2;
;  989          1388  4            END
;  990          1389  3            ELSE I = .I + 1;
;  991          1390  3
;  992          1391  2    END;
;  993          1392  2 RETURN 1;
;  994          1393  1 END;
```

```
                                01FC 00000 FORMAT_VOL_OWNER:
                                           .WORD   Save R2,R3,R4,R5,R6,R7,R8                                   ; 1344
                          58    0000'  CF  9E 00002         MOVAB   DATA_INDEX, R8
        0E          20    6E           00  2C 00007         MOVC5   #0, (SP), #32, #14, VOL_OWNER               ; 1354
                            D0    A8       0000C
                                  56  7C 0000E         CLRQ    I                                               ; 1356
                          50    14    BC  90 00010         MOVB    @20(CONTEXT), TEMP                          ; 1357
                          2C          50  91 00014         CMPB    TEMP, #44                                   ; 1358
                                      10  13 00017         BEQL    1$
                          50    E0    A8  9E 00019         MOVAB   DATA_PTR, R0                                ; 1362
              00 B840     14    BC    10  AC 28 0001D      MOVC3   16(CONTEXT), @20(CONTEXT), @DATA_INDEX[R0]
                          68    10    AC  C0 00025         ADDL2   16(CONTEXT), DATA_INDEX                     ; 1363
                50        68          01  C3 00029 1$:     SUBL3   #1, DATA_INDEX, R0                          ; 1365
                          50          56  D1 0002D         CMPL    I, R0
                                      7D  14 00030         BGTR    11$
                          52    E0 A846   90 00032         MOVB    DATA_PTR[I], CHAR                           ; 1368
                    41    8F          52  91 00037         CMPB    CHAR, #65                                   ; 1371
                                      06  1F 0003B         BLSSU   2$
                    5A    8F          52  91 0003D         CMPB    CHAR, #90
                                      38  1B 00041         BLEQU   6$
                          20          52  91 00043 2$:     CMPB    CHAR, #32                                   ; 1372
                                      05  1F 00046         BLSSU   3$
                          22          52  91 00048         CMPB    CHAR, #34
                                      2E  1B 0004B         BLEQU   6$
                          25          52  91 0004D 3$:     CMPB    CHAR, #37                                   ; 1373
                                      05  1F 00050         BLSSU   4$
                          3F          52  91 00052         CMPB    CHAR, #63
                                      24  1B 00055         BLEQU   6$
                    5F    8F          52  91 00057 4$:     CMPB    CHAR, #95                                   ; 1374
                                      1E  13 0005B         BEQL    6$
                    61    8F          52  91 0005D         CMPB    CHAR, #97                                   ; 1375
                                      0B  1F 00061         BLSSU   5$
                    7A    8F          52  91 00063         CMPB    CHAR, #122
                                      05  1A 00067         BGTRU   5$
                          52          20  82 00069         SUBB2   #32, CHAR
                                      0D  11 0006C         BRB     6$
                              00758114  8F  DD 0006E 5$:     PUSHL   #7700756                                    ; 1376
        00000000G     00              01  FB 00074         CALLS   #1, LIB$STOP
                                      51  D4 0007B 6$:     CLRL    R1                                          ; 1378
                          22          52  91 0007D         CMPB    CHAR, #34
                                      0B  12 00080         BNEQ    7$
                                      51  D6 00082         INCL    R1
        50                68          01  C3 00084         SUBL3   #1, DATA_INDEX, R0
                          50          56  D1 00088         CMPL    I, R0
                                      22  13 0008B         BEQL    11$
                          04          51  E9 0008D 7$:     BLBC    R1, 8$                                      ; 1380
                                      56  D5 00090         TSTL    I
                                      16  13 00092         BEQL    9$
                    D0 A847           52  90 00094 8$:     MOVB    CHAR, VOL_OWNER[J]                          ; 1383
                                      57  D6 00099         INCL    J                                           ; 1384
                          0C          51  E9 0009B         BLBC    R1, 9$                                      ; 1385
                          22    E1 A846   91 0009E         CMPB    DATA_PTR+1[I], #34
                                      05  12 000A3         BNEQ    9$
                          56          02  C0 000A5         ADDL2   #2, I                                       ; 1387
                                      02  11 000A8         BRB     10$                                         ; 1380
                                      56  D6 000AA 9$:     INCL    I                                           ; 1389
```

```
                          FF7A  31 000AC 10$:     BRW     1$                              ; 1365
                50          01  D0 000AF 11$:     MOVL    #1, R0                          ; 1392
                                04 000B2         RET                                     ; 1393

; Routine Size:  179 bytes,    Routine Base:  $CODE$ + 073C
```

```
;  996          1394  1 ROUTINE MOVE_CHAR=
;  997          1395  2 BEGIN
;  998          1396  2
;  999          1397  2 LOCAL
; 1000          1398  2         TEMP_DATA_INDEX;
; 1001          1399  2
; 1002          1400  2 TPARSE_ARGS (CONTEXT);
; 1003          1401  2 TEMP_DATA_INDEX = .DATA_INDEX + .CONTEXT[TPA$L_TOKENCNT];
; 1004          1402  2 IF .TEMP_DATA_INDEX GTR 32
; 1005          1403  2 THEN
; 1006          1404  2     ERR_EXIT(INIT$_BADOWNID);
; 1007          1405  2 CH$MOVE (.CONTEXT[TPA$L_TOKENCNT], .CONTEXT[TPA$L_TOKENPTR], DATA_PTR[.DATA_INDEX]);
; 1008          1406  2 DATA_INDEX = .TEMP_DATA_INDEX;
; 1009          1407  2 RETURN 1;
; 1010          1408  1 END;


                                        00FC 00000 MOVE_CHAR:
                                                   .WORD   Save R2,R3,R4,R5,R6,R7                          ; 1394
                          57     0000'  CF  9E 00002       MOVAB   DATA_INDEX, R7                          ;
                    56    67       10  AC  C1 00007       ADDL3   16(CONTEXT), DATA_INDEX, TEMP_DATA_INDEX ; 1401
                          20           56  D1 0000C       CMPL    TEMP_DATA_INDEX, #32                    ; 1402
                                       0D  15 0000F       BLEQ    1$                                      ;
                             00758114  8F  DD 00011       PUSHL   #7700756                                ; 1404
                00000000G    00        01  FB 00017       CALLS   #1, LIB$STOP                            ;
                          50     E0    A7  9E 0001E 1$:    MOVAB   DATA_PTR, R0                            ; 1405
                00 B740         14  BC       10  AC  28 00022       MOVC3   16(CONTEXT), @20(CONTEXT), @DATA_INDEX[R0]
                          67           56  D0 0002A       MOVL    TEMP_DATA_INDEX, DATA_INDEX             ; 1406
                          50           01  D0 0002D       MOVL    #1, R0                                  ; 1407
                                       04 00030           RET                                             ; 1408

; Routine Size:  49 bytes,    Routine Base:  $CODE$ + 07EF
```

```
;  1012          1409  1 !+
;  1013          1410  1 !
;  1014          1411  1 ! TPARSE state tables to parse the various qualifier value strings.
;  1015          1412  1 !
;  1016          1413  1 !-
;  1017          1414  1
;  1018          1415  1 !
;  1019          1416  1 ! Parse magtape density (either "800", "1600", or "6250").
;  1020          1417  1 !  Also for floppy disks only (either "SINGLE" or "DOUBLE").
;  1021          1418  1 !
;  1022          1419  1 $INIT_STATE (DENSITY_STB, DENSITY_KTB);
;  1023          1420  1
;  1024        P 1421  1 $STATE  (,
;  1025        P 1422  1          ('6250',,, 1^(OPT_DENS_6250 - 32), INIT_OPTIONS + 4),
;  1026        P 1423  1          ('1600',,, 1^(OPT_DENS_1600 - 32), INIT_OPTIONS + 4),
;  1027        P 1424  1          ('800',,, 1^OPT_DENS_800, INIT_OPTIONS),
;  1028        P 1425  1          ('SINGLE',,, 1^(OPT_DENS_SING - 32), INIT_OPTIONS + 4),
;  1029        P 1426  1          ('DOUBLE',,, 1^(OPT_DENS_DOUB - 32), INIT_OPTIONS + 4)
;  1030          1427  1          );
;  1031          1428  1
;  1032        P 1429  1 $STATE  (,
;  1033        P 1430  1          (TPA$_EOS, TPA$_EXIT)
;  1034          1431  1          );
;  1035          1432  1 !
;  1036          1433  1 ! Parse disk structure level (either "1" or "2").
;  1037          1434  1 !
;  1038          1435  1 $INIT_STATE (STRUCTURE_STB, STRUCTURE_KTB);
;  1039          1436  1
;  1040        P 1437  1 $STATE  (,
;  1041        P 1438  1          ('1',,, 1^OPT_STRUCTURE1, INIT_OPTIONS),
;  1042        P 1439  1          ('2')
;  1043          1440  1          );
;  1044          1441  1
;  1045        P 1442  1 $STATE  (,
;  1046        P 1443  1          (TPA$_EOS, TPA$_EXIT)
;  1047          1444  1          );
;  1048          1445  1 !
;  1049          1446  1 ! PARSE OVERRIDE QUALIFIERS (EXPIRATION)
;  1050          1447  1 !
;  1051          1448  1 $INIT_STATE(OVERRIDE_STB, OVERRIDE_KTB);
;  1052          1449  1
;  1053        P 1450  1 $STATE  (NEXTOVR,
;  1054        P 1451  1          ('EXPIRATION',,,1^OPT_OVR_EXP,INIT_OPTIONS),
;  1055        P 1452  1          ('ACCESSIBILITY',,,1^OPT_OVR_ACC,INIT_OPTIONS),
;  1056        P 1453  1          ('OWNER_IDENTIFIER',,,1^(OPT_OVR_VOLO-32),INIT_OPTIONS+4)
;  1057          1454  1          );
;  1058          1455  1
;  1059          1456  1
;  1060        P 1457  1 $STATE  (,
;  1061        P 1458  1          (',',NEXTOVR),
;  1062        P 1459  1          (TPA$_EOS,TPA$_EXIT)
;  1063          1460  1          );
;  1064          1461  1 !
;  1065          1462  1 ! Parse protection string "(SYSTEM:RWED,OWNER:RWED,GROUP:RWED,WORLD:RWED)"
;  1066          1463  1 !
;  1067          1464  1 $INIT_STATE (PROTECTION_STB, PROTECTION_KTB);
;  1068          1465  1
```

```
;  1069          P 1466  1 $STATE  (NEXTPRO,
;  1070          P 1467  1          ('SYSTEM', SYPR,, %X'000F0000', PROT_VAL),
;  1071          P 1468  1          ('OWNER',  OWPR,, %X'00F00000', PROT_VAL),
;  1072          P 1469  1          ('GROUP',  GRPR,, %X'0F000000', PROT_VAL),
;  1073          P 1470  1          ('WORLD',  WOPR,, %X'F0000000', PROT_VAL)
;  1074            1471  1          );
;  1075            1472  1
;  1076          P 1473  1 $STATE  (SYPR,
;  1077          P 1474  1          (':'),
;  1078          P 1475  1          ('='),
;  1079          P 1476  1          (TPA$_LAMBDA, ENDPRO)
;  1080            1477  1          );
;  1081            1478  1
;  1082          P 1479  1 $STATE  (SYPRO,
;  1083          P 1480  1          ('R', SYPRO,, %X'0001', PROT_VAL),
;  1084          P 1481  1          ('W', SYPRO,, %X'0002', PROT_VAL),
;  1085          P 1482  1          ('E', SYPRO,, %X'0004', PROT_VAL),
;  1086          P 1483  1          ('D', SYPRO,, %X'0008', PROT_VAL),
;  1087          P 1484  1          (TPA$_LAMBDA, ENDPRO)
;  1088            1485  1          );
;  1089            1486  1
;  1090          P 1487  1 $STATE  (OWPR,
;  1091          P 1488  1          (':'),
;  1092          P 1489  1          ('='),
;  1093          P 1490  1          (TPA$_LAMBDA, ENDPRO)
;  1094            1491  1          );
;  1095            1492  1
;  1096          P 1493  1 $STATE  (OWPRO,
;  1097          P 1494  1          ('R', OWPRO,, %X'0010', PROT_VAL),
;  1098          P 1495  1          ('W', OWPRO,, %X'0020', PROT_VAL),
;  1099          P 1496  1          ('E', OWPRO,, %X'0040', PROT_VAL),
;  1100          P 1497  1          ('D', OWPRO,, %X'0080', PROT_VAL),
;  1101          P 1498  1          (TPA$_LAMBDA, ENDPRO)
;  1102            1499  1          );
;  1103            1500  1
;  1104          P 1501  1 $STATE  (GRPR,
;  1105          P 1502  1          (':'),
;  1106          P 1503  1          ('='),
;  1107          P 1504  1          (TPA$_LAMBDA, ENDPRO)
;  1108            1505  1          );
;  1109            1506  1
;  1110          P 1507  1 $STATE  (GRPRO,
;  1111          P 1508  1          ('R', GRPRO,, %X'0100', PROT_VAL),
;  1112          P 1509  1          ('W', GRPRO,, %X'0200', PROT_VAL),
;  1113          P 1510  1          ('E', GRPRO,, %X'0400', PROT_VAL),
;  1114          P 1511  1          ('D', GRPRO,, %X'0800', PROT_VAL),
;  1115          P 1512  1          (TPA$_LAMBDA, ENDPRO)
;  1116            1513  1          );
;  1117            1514  1
;  1118          P 1515  1 $STATE  (WOPR,
;  1119          P 1516  1          (':'),
;  1120          P 1517  1          ('='),
;  1121          P 1518  1          (TPA$_LAMBDA, ENDPRO)
;  1122            1519  1          );
;  1123            1520  1
;  1124          P 1521  1 $STATE  (WOPRO,
;  1125          P 1522  1          ('R', WOPRO,, %X'1000', PROT_VAL),
```

```
;  1126        P 1523  1          ('W', WOPRO,, %X'2000', PROT_VAL),
;  1127        P 1524  1          ('E', WOPRO,, %X'4000', PROT_VAL),
;  1128        P 1525  1          ('D', WOPRO,, %X'8000', PROT_VAL),
;  1129        P 1526  1          (TPA$_LAMBDA, ENDPRO)
;  1130          1527  1          );
;  1131          1528  1
;  1132        P 1529  1 $STATE  (ENDPRO,
;  1133        P 1530  1          (',', NEXTPRO),
;  1134        P 1531  1          (TPA$_EOS, TPA$_EXIT)
;  1135          1532  1          );
;  1136          1533  1
;  1137          1534  1 !
;  1138          1535  1 ! Parse UIC string and store binary value.
;  1139          1536  1 !
;  1140          1537  1 $INIT_STATE (UIC_STB, UIC_KTB);
;  1141          1538  1
;  1142        P 1539  1 $STATE  (,
;  1143        P 1540  1          (TPA$_IDENT,,,OWNER_UIC)
;  1144          1541  1          );
;  1145          1542  1
;  1146        P 1543  1 $STATE  (,
;  1147        P 1544  1          (TPA$_EOS, TPA$_EXIT)
;  1148          1545  1          );
;  1149          1546  1
;  1150          1547  1 !
;  1151          1548  1 ! Parse INDEX options (BEGINNING, MIDDLE, END, OR n)
;  1152          1549  1 !
;  1153          1550  1 $INIT_STATE (INDEX_STB, INDEX_KTB);
;  1154          1551  1
;  1155        P 1552  1 $STATE  (,
;  1156        P 1553  1          ('BEGINNING',,, 1^OPT_INDEX_BEG, INIT_OPTIONS),
;  1157        P 1554  1          ('MIDDLE',,, 1^OPT_INDEX_MID, INIT_OPTIONS),
;  1158        P 1555  1          ('END',,, 1^OPT_INDEX_END, INIT_OPTIONS),
;  1159        P 1556  1          (TPA$_DECIMAL,, GET_INDEX_LBN, T^OPT_INDEX_LBN, INIT_OPTIONS)
;  1160          1557  1          );
;  1161          1558  1
;  1162        P 1559  1 $STATE  (,
;  1163        P 1560  1          (TPA$_EOS, TPA$_EXIT)
;  1164          1561  1          );
;  1165          1562  1 !
;  1166          1563  1 ! Parse data check options, of the form [READ][,WRITE]. Default is write.
;  1167          1564  1 !
;  1168          1565  1 $INIT_STATE (DATACHECK_STB, DATACHECK_KTB);
;  1169          1566  1
;  1170        P 1567  1 $STATE  (,
;  1171        P 1568  1          (TPA$_EOS, TPA$_EXIT,, 1^OPT_WRITECHECK, INIT_OPTIONS),
;  1172        P 1569  1          (TPA$_LAMBDA)
;  1173          1570  1          );
;  1174          1571  1
;  1175        P 1572  1 $STATE  (CHECKOPT,
;  1176        P 1573  1          ('READ',,, 1^OPT_READCHECK, INIT_OPTIONS),
;  1177        P 1574  1          ('WRITE',,, 1^OPT_WRITECHECK, INIT_OPTIONS)
;  1178          1575  1          );
;  1179          1576  1
;  1180        P 1577  1 $STATE  (,
;  1181        P 1578  1          (',', CHECKOPT),
;  1182        P 1579  1          (TPA$_EOS, TPA$_EXIT)
```

```
;  1183          1580  1          );
;  1184          1581  1
;  1185          1582  1 !
;  1186          1583  1 ! Parse bad block data, consisting of entries of the form LBN:count or
;  1187          1584  1 ! sector.track.cylinder:count, separated by commas.
;  1188          1585  1 !
;  1189          1586  1 $INIT_STATE (BADBLOCKS_STB, BADBLOCKS_KTB);
;  1190          1587  1
;  1191        P 1588  1 $STATE  (NEXTBLK,
;  1192        P 1589  1          (TPA$_DECIMAL,, GET_BAD_LBN)
;  1193          1590  1          );
;  1194          1591  1
;  1195        P 1592  1 $STATE  (,
;  1196        P 1593  1          ('.', TRACK)
;  1197        P 1594  1          (TPA$_LAMBDA)
;  1198          1595  1          );
;  1199          1596  1
;  1200        P 1597  1 $STATE  (COLON,
;  1201        P 1598  1          (':', BLKCNT),
;  1202        P 1599  1          (TPA$_LAMBDA)
;  1203          1600  1          );
;  1204          1601  1
;  1205        P 1602  1 $STATE  (BLKEND,
;  1206        P 1603  1          (',', NEXTBLK),
;  1207        P 1604  1          (TPA$_EOS, TPA$_EXIT)
;  1208          1605  1          );
;  1209          1606  1
;  1210        P 1607  1 $STATE  (BLKCNT,
;  1211        P 1608  1          (TPA$_DECIMAL, BLKEND, GET_BAD_COUNT)
;  1212          1609  1          );
;  1213          1610  1
;  1214        P 1611  1 $STATE  (TRACK,
;  1215        P 1612  1          (TPA$_DECIMAL,, GET_BAD_TRACK)
;  1216          1613  1          );
;  1217          1614  1
;  1218        P 1615  1 $STATE  (,
;  1219        P 1616  1          ('.')
;  1220          1617  1          );
;  1221          1618  1
;  1222        P 1619  1 $STATE  (,
;  1223        P 1620  1          (TPA$_DECIMAL, COLON, GET_BAD_CYL)
;  1224          1621  1          );
;  1225          1622  1
;  1226          1623  1 !
;  1227          1624  1 ! Parse LABEL option VOLUME_ACCESSIBILITY = "x" and OWNER_IDENTIFIER = "vol_owner"
;  1228          1625  1 !
;  1229          1626  1 $INIT_STATE (LABEL_QUAL_STB, LABEL_QUAL_KTB);
;  1230          1627  1
;  1231        P 1628  1 $STATE  (LABEL_QUAL,
;  1232        P 1629  1          ('VOLUME_ACCESSIBILITY',VOLUME,, 1^(OPT_LABEL_QUAL-32), INIT_OPTIONS+4),
;  1233        P 1630  1          ('OWNER_IDENTIFIER',OWNER,, 1^(OPT_VOL_OWNER-32), INIT_OPTIONS+4)
;  1234          1631  1          );
;  1235          1632  1
;  1236        P 1633  1 $STATE  (VOLUME,
;  1237        P 1634  1          (':',VALUEVOLACC),
;  1238        P 1635  1          ('=',VALUEVOLACC)
;  1239          1636  1          );
```

```
;  1240          1637  1
;  1241        P 1638  1 $STATE  (VALUEVOLACC,
;  1242        P 1639  1          (TPA$_ANY,DONE,GET_VOL_ACC)
;  1243          1640  1          );
;  1244          1641  1
;  1245        P 1642  1 $STATE  (OWNER,
;  1246        P 1643  1          (':',VALUEOWNER,),
;  1247        P 1644  1          ('=',VALUEOWNER,)
;  1248          1645  1          );
;  1249          1646  1
;  1250        P 1647  1 $STATE  (VALUEOWNER,
;  1251        P 1648  1          ('"',ANSI_VOLO,MOVE_CHAR),
;  1252        P 1649  1          (',',LABEL_QUAL,FORMAT_VOL_OWNER),
;  1253        P 1650  1          (TPA$_SYMBOL,DONE,FORMAT_VOL_OWNER),
;  1254        P 1651  1          (TPA$_LAMBDA,DONE,FORMAT_VOL_OWNER)
;  1255          1652  1          );
;  1256          1653  1
;  1257        P 1654  1 $STATE  (ANSI_VOLO,
;  1258        P 1655  1          ('"',VALUEOWNER,MOVE_CHAR),
;  1259        P 1656  1          (TPA$_ANY,ANSI_VOLO,MOVE_CHAR),
;  1260        P 1657  1          (TPA$_EOS,TPA$_EXIT)
;  1261          1658  1          );
;  1262          1659  1
;  1263        P 1660  1 $STATE  (DONE,
;  1264        P 1661  1          (',',LABEL_QUAL),
;  1265        P 1662  1          (TPA$_EOS,TPA$_EXIT)
;  1266          1663  1          );
;  1267          1664  1
;  1268          1665  1 END
;  1269          1666  0 ELUDOM


                                                .PSECT  _LIB$KEY1$,NOWRT,  SHR,  PIC,1

                                        00000 ;TPA$KEYSTO
                                              U.2:      .BLKB   0
                                30  35  32  36  00000 ;TPA$KEYST
                                              U.4:      .ASCII  \6250\                  ;
                                    FF  00004           .BYTE   -1                      ;
                                        00005 ;TPA$KEYSTO
                                              U.8:      .BLKB   C
                                30  30  36  31  00005 ;TPA$KEYST
                                              U.10:     .ASCII  \1600\                  ;
                                    FF  00009           .BYTE   -1                      ;
                                        0000A ;TPA$KEYSTO
                                              U.14:     .BLKB   0
                                    30  30  38  0000A ;TPA$KEYST
                                              U.16:     .ASCII  \800\                   ;
                                    FF  0000D           .BYTE   -1                      ;
                                        0000E ;TPA$KEYSTO
                                              U.20:     .BLKB   0
                            45  4C  47  4E  49  53  0000E ;TPA$KEYST
                                              U.22:     .ASCII  \SINGLE\                ;
                                    FF  00014           .BYTE   -1                      ;
                                        00015 ;TPA$KEYSTO
                                              U.26:     .BLKB   0
```

```
                                                45 4C 42 55 4F 44  00015 ;TPA$KEYST
                                                                   U.28:    .ASCII  \DOUBLE\
                                                FF                 0001B            .BYTE   -1
                                                FF                 0001C ;TPA$KEYFILL
                                                                   U.32:    .BYTE   -1
                                                                   0001D ;TPA$KEYSTO
                                                                   U.43:    .BLKB   0
                                  4E 4F 49 54 41 52 49 50 58 45    0001D ;TPA$KEYST
                                                                   U.45:    .ASCII  \EXPIRATION\
                                                FF                 00027            .BYTE   -1
                                                                   00028 ;TPA$KEYSTO
                                                                   U.49:    .BLKB   0
                     59 54 49 4C 49 42 49 53 53 45 43 43 41        00028 ;TPA$KEYST
                                                                   U.51:    .ASCII  \ACCESSIBILITY\
                                                FF                 00035            .BYTE   -1
                                                                   00036 ;TPA$KEYSTO
                                                                   U.55:    .BLKB   0
45 49 46 49 54 4E 45 44 49 5F 52 45 4E 57 4F                       00036 ;TPA$KEYST
                                                                   U.57:    .ASCII  \OWNER_IDENTIFIER\
                                                52                 00045
                                                FF                 00046            .BYTE   -1
                                                FF                 00047 ;TPA$KEYFILL
                                                                   U.61:    .BYTE   -1
                                                                   00048 ;TPA$KEYSTO
                                                                   U.67:    .BLKB   0
                                                4D 45 54 53 59 53  00048 ;TPA$KEYST
                                                                   U.69:    .ASCII  \SYSTEM\
                                                FF                 0004E            .BYTE   -1
                                                                   0004F ;TPA$KEYSTO
                                                                   U.75:    .BLKB   0
                                     52 45 4E 57 4F                0004F ;TPA$KEYST
                                                                   U.77:    .ASCII  \OWNER\
                                                FF                 00054            .BYTE   -1
                                                                   00055 ;TPA$KEYSTO
                                                                   U.83:    .BLKB   0
                                     50 55 4F 52 47                00055 ;TPA$KEYST
                                                                   U.85:    .ASCII  \GROUP\
                                                FF                 0005A            .BYTE   -1
                                                                   0005B ;TPA$KEYSTO
                                                                   U.91:    .BLKB   0
                                     44 4C 52 4F 57                0005B ;TPA$KEYST
                                                                   U.93:    .ASCII  \WORLD\
                                                FF                 00060            .BYTE   -1
                                                FF                 00061 ;TPA$KEYFILL
                                                                   U.99:    .BYTE   -1
                                                                   00062 ;TPA$KEYSTO
                                                                   U.199:   .BLKB   0
                  47 4E 49 4E 4E 49 47 45 42                       00062 ;TPA$KEYST
                                                                   U.201:   .ASCII  \BEGINNING\
                                                FF                 0006B            .BYTE   -1
                                                                   0006C ;TPA$KEYSTO
                                                                   U.205:   .BLKB   0
                                  45 4C 44 44 49 4D                0006C ;TPA$KEYST
                                                                   U.207:   .ASCII  \MIDDLE\
                                                1F                 00072            .BYTE   -1
                                                                   00073 ;TPA$KEYSTO
                                                                   U.211:   .BLKB   0
```

```
                                                        44  4E  45  00073 ;TPA$KEYST
                                                                        U.213:  .ASCII  \END\                                   ;
                                                            FF  00076           .BYTE   -1                                  ;
                                                            FF  00077 ;TPA$KEYFILL
                                                                        U.221:  .BYTE   -1                                  ;
                                                                00078 ;TPA$KEYSTO
                                                                        U.230:  .BLKB   0
                                                    44  41  45  52  00078 ;TPA$KEYST
                                                                        U.232:  .ASCII  \READ\                              ;
                                                            FF  0007C           .BYTE   -1                                  ;
                                                                0007D ;TPA$KEYSTO
                                                                        U.236:  .BLKB   0
                                                45  54  49  52  57  0007D ;TPA$KEYST
                                                                        U.238:  .ASCII  \WRITE\                             ;
                                                            FF  00082           .BYTE   -1                                  ;
                                                            FF  00083 ;TPA$KEYFILL
                                                                        U.242:  .BYTE   -1                                  ;
                                                                00084 ;TPA$KEYSTO
                                                                        U.272:  .BLKB   0
42  49  53  53  45  43  43  41  5F  45  4D  55  4C  4F  56  00084 ;TPA$KEYST
                                                                        U.274:  .ASCII  \VOLUME_ACCESSIBILITY\                  ;
                                                59  54  49  4C  49  00093
                                                            FF  00098           .BYTE   -1                                  ;
                                                                00099 ;TPA$KEYSTO
                                                                        U.280:  .BLKB   0
45  49  46  49  54  4E  45  44  49  5F  52  45  4E  57  4F  00099 ;TPA$KEYST
                                                                        U.282:  .ASCII  \OWNER_IDENTIFIER\                      ;
                                                            52  000A8
                                                            FF  000A9           .BYTE   -1                                  ;
                                                            FF  000AA ;TPA$KEYFILL
                                                                        U.288:  .BYTE   -1                                  ;

                                                                        .PSECT  _LIB$STATE$,NOWRT,  SHR,  PIC,1

                                                                00000 DENSITY_STB::
                                                                        .BLKB   0
                                                             6100  00000 ;TPA$TYPE
                                                                        U.5:    .WORD   24832                               ;
                                                        00000000*   00002 ;TPA$ADDR
                                                                        U.6:    .LONG   <<<INIT_OPTIONS+4>-U.6>-4>          ;
                                                        00000008    00006 ;TPA$MASK
                                                                        U.7:    .LONG   8                                   ;
                                                             6101  0000A ;TPA$TYPE
                                                                        U.11:   .WORD   24833                               ;
                                                        00000000*   0000C ;TPA$ADDR
                                                                        U.12:   .LONG   <<<INIT_OPTIONS+4>-U.12>-4>         ;
                                                        00000002    00010 ;TPA$MASK
                                                                        U.13:   .LONG   2                                   ;
                                                             6102  00014 ;TPA$TYPE
                                                                        U.17:   .WORD   24834                               ;
                                                        00000000*   00016 ;TPA$ADDR
                                                                        U.18:   .LONG   <<INIT_OPTIONS-U.18>-4>             ;
                                                        00000002    0001A ;TPA$MASK
                                                                        U.19:   .LONG   2                                   ;
                                                             6103  0001E ;TPA$TYPE
                                                                        U.23:   .WORD   24835                               ;
                                                        00000000*   00020 ;TPA$ADDR
```

```
                        U.24:   .LONG   <<<INIT_OPTIONS+4>-U.24>-4>                     ;
00000010  00024 ;TPA$MASK
                        U.25:   .LONG   16                                              ;
    6504  00028 ;TPA$TYPE
                        U.29:   .WORD   25860                                           ;
00000000* 0002A ;TPA$ADDR
                        U.30:   .LONG   <<<INIT_OPTIONS+4>-U.30>-4>                     ;
00000020  0002E ;TPA$MASK
                        U.31:   .LONG   32                                              ;
    15F7  00032 ;TPA$TYPE
                        U.33:   .WORD   5623                                            ;
    FFFF  00034 ;TPA$TARGET
                        U.34:   .WORD   -1                                              ;
          00036         .BLKB   2
          00038 STRUCTURE_STB::
                        .BLKB   0
    6031  00038 ;TPA$TYPE
                        U.36:   .WORD   24625                                           ;
00000000* 0003A ;TPA$ADDR
                        U.37:   .LONG   <<INIT_OPTIONS-U.37>-4>                         ;
80000000  0003E ;TPA$MASK
                        U.38:   .LONG   -2147483648                                     ;
    0432  00042 ;TPA$TYPE
                        U.39:   .WORD   1074                                            ;
    15F7  00044 ;TPA$TYPE
                        U.40:   .WORD   5623                                            ;
    FFFF  00046 ;TPA$TARGET
                        U.41:   .WORD   -1                                              ;
          00048 OVERRIDE_STB::
                        .BLKB   0
          00048 NEXTOVR:.BLKB   0
    6100  00048 ;TPA$TYPE
                        U.46:   .WORD   24832                                           ;
00000000* 0004A ;TPA$ADDR
                        U.47:   .LONG   <<INIT_OPTIONS-U.47>-4>                         ;
08000000  0004E ;TPA$MASK
                        U.48:   .LONG   134217728                                       ;
    6101  00052 ;TPA$TYPE
                        U.52:   .WORD   24833                                           ;
00000000* 00054 ;TPA$ADDR
                        U.53:   .LONG   <<INIT_OPTIONS-U.53>-4>                         ;
40000000  00058 ;TPA$MASK
                        U.54:   .LONG   1073741824                                      ;
    6502  0005C ;TPA$TYPE
                        U.58:   .WORD   25858                                           ;
00000000* 0005E ;TPA$ADDR
                        U.59:   .LONG   <<<INIT_OPTIONS+4>-U.59>-4>                     ;
00000100  00062 ;TPA$MASK
                        U.60:   .LONG   256                                             ;
    102C  00066 ;TPA$TYPE
                        U.62:   .WORD   4140                                            ;
    0000* 00068 ;TPA$TARGET
                        U.63:   .WORD   <<NEXTOVR-U.63>-2>                              ;
    15F7  0006A ;TPA$TYPE
                        U.64:   .WORD   5623                                            ;
    FFFF  0006C ;TPA$TARGET
                        U.65:   .WORD   -1                                              ;
```

```
                00006E          .BLKB   2
                00070 PROTECTION_STB::
                                .BLKB   0
                00070 NEXTPRO:  .BLKB   0
     7100       00070 ;TPA$TYPE
                      U.70:     .WORD   28928                                   ;
00000000*       00072 ;TPA$ADDR
                      U.71:     .LONG   <<PROT_VAL-U.71>-4>                     ;
000F0000        00076 ;TPA$MASK
                      U.72:     .LONG   983040                                  ;
     0000*      0007A ;TPA$TARGET
                      U.74:     .WORD   <<U.73-U.74>-2>                         ;
     7101       0007C ;TPA$TYPE
                      U.78:     .WORD   28929                                   ;
00000000*       0007E ;TPA$ADDR
                      U.79:     .LONG   <<PROT_VAL-U.79>-4>                     ;
00F00000        00082 ;TPA$MASK
                      U.80:     .LONG   15728640                                ;
     0000*      00086 ;TPA$TARGET
                      U.82:     .WORD   <<U.81-U.82>-2>                         ;
     7102       00088 ;TPA$TYPE
                      U.86:     .WORD   28930                                   ;
00000000*       0008A ;TPA$ADDR
                      U.87:     .LONG   <<PROT_VAL-U.87>-4>                     ;
0F000000        0008E ;TPA$MASK
                      U.88:     .LONG   251658240                               ;
     0000*      00092 ;TPA$TARGET
                      U.90:     .WORD   <<U.89-U.90>-2>                         ;
     7503       00094 ;TPA$TYPE
                      U.94:     .WORD   29955                                   ;
00000000*       00096 ;TPA$ADDR
                      U.95:     .LONG   <<PROT_VAL-U.95>-4>                     ;
F0000000        0009A ;TPA$MASK
                      U.96:     .LONG   -268435456                              ;
     0000*      0009E ;TPA$TARGET
                      U.98:     .WORD   <<U.97-U.98>-2>                         ;
                000A0 ;SYPR
                      U.73:     .BLKB   0
     003A       000A0 ;TPA$TYPE
                      U.100:    .WORD   58                                      ;
     003D       000A2 ;TPA$TYPE
                      U.101:    .WORD   61                                      ;
     15F6       000A4 ;TPA$TYPE
                      U.102:    .WORD   5622                                    ;
     0000*      000A6 ;TPA$TARGET
                      U.104:    .WORD   <<U.103-U.104>-2>                       ;
                000A8 SYPRO:    .BLKB   0
     7052       000A8 ;TPA$TYPE
                      U.105:    .WORD   28754                                   ;
00000000*       000AA ;TPA$ADDR
                      U.106:    .LONG   <<PROT_VAL-U.106>-4>                    ;
00000001        000AE ;TPA$MASK
                      U.107:    .LONG   1                                       ;
     0000*      000B2 ;TPA$TARGET
                      U.108:    .WORD   <<SYPRO-U.108>-2>                       .
     7057       000B4 ;TPA$TYPE
                      U.109:    .WORD   28759                                   ;
```

```
00000000* 000B6 ;TPA$ADDR
                U.110:  .LONG   <<PROT_VAL-U.110>-4>                    ;
00000002  000BA ;TPA$MASK
                U.111:  .LONG   2                                       ;
    0000* 000BE ;TPA$TARGET
                U.112:  .WORD   <<SYPRO-U.112>-2>                       ;
    7045  000C0 ;TPA$TYPE
                U.113:  .WORD   28741                                   ;
00000000* 000C2 ;TPA$ADDR
                U.114:  .LONG   <<PROT_VAL-U.114>-4>                    ;
00000004  000C6 ;TPA$MASK
                U.115:  .LONG   4                                       ;
    0000* 000CA ;TPA$TARGET
                U.116:  .WORD   <<SYPRO-U.116>-2>                       ;
    7044  000CC ;TPA$TYPE
                U.117:  .WORD   28740                                   ;
00000000* 000CE ;TPA$ADDR
                U.118:  .LONG   <<PROT_VAL-U.118>-4>                    ;
00000008  000D2 ;TPA$MASK
                U.119:  .LONG   8                                       ;
    0000* 000D6 ;TPA$TARGET
                U.120:  .WORD   <<SYPRO-U.120>-2>                       ;
    15F6  000D8 ;TPA$TYPE
                U.121:  .WORD   5622                                    ;
    0000* 000DA ;TPA$TARGET
                U.122:  .WORD   <<U.103-U.122>-2>                       ;
          000DC ;OWPR
                U.81:   .BLKB   0
    003A  000DC ;TPA$TYPE
                U.123:  .WORD   58                                      ;
    003D  000DE ;TPA$TYPE
                U.124:  .WORD   61                                      ;
    15F6  000E0 ;TPA$TYPE
                U.125:  .WORD   5622                                    ;
    0000* 000E2 ;TPA$TARGET
                U.126:  .WORD   <<U.103-U.126>-2>                       ;
          000E4 OWPRO:  .BLKB   0
    7052  000E4 ;TPA$TYPE
                U.127:  .WORD   28754                                   ;
00000000* 000E6 ;TPA$ADDR
                U.128:  .LONG   <<PROT_VAL-U.128>-4>                    ;
00000010  000EA ;TPA$MASK
                U.129:  .LONG   16                                      ;
    0000* 000EE ;TPA$TARGET
                U.130:  .WORD   <<OWPRO-U.130>-2>                       ;
    7057  000F0 ;TPA$TYPE
                U.131:  .WORD   28759                                   ;
00000000* 000F2 ;TPA$ADDR
                U.132:  .LONG   <<PROT_VAL-U.132>-4>                    ;
00000020  000F6 ;TPA$MASK
                U.133:  .LONG   32                                      ;
    0000* 000FA ;TPA$TARGET
                U.134:  .WORD   <<OWPRO-U.134>-2>                       ;
    7045  000FC ;TPA$TYPE
                U.135:  .WORD   28741                                   ;
00000000* 000FE ;TPA$ADDR
                U.136:  .LONG   <<PROT_VAL-U.136>-4>                    ;
```

```
00000040  00102 ;TPA$MASK
                U.137:  .LONG   64                                      ;
    0000* 00106 ;TPA$TARGET
                U.138:  .WORD   <<OWPRO-U.138>-2>                       ;
    7044  00108 ;TPA$TYPE
                U.139:  .WORD   28740                                   ;
00000000* 0010A ;TPA$ADDR
                U.140:  .LONG   <<PROT_VAL-U.140>-4>                    ;
00000080  0010E ;TPA$MASK
                U.141:  .LONG   128                                     ;
    0000* 00112 ;TPA$TARGET
                U.142:  .WORD   <<OWPRO-U.142>-2>                       ;
    15F6  00114 ;TPA$TYPE
                U.143:  .WORD   5622                                    ;
    0000* 00116 ;TPA$TARGET
                U.144:  .WORD   <<U.103-U.144>-2>                       ;
          00118 ;GRPR
                U.89:   .BLKB   0
    003A  00118 ;TPA$TYPE
                U.145:  .WORD   58                                      ;
    003D  0011A ;TPA$TYPE
                U.146:  .WORD   61                                      ;
    15F6  0011C ;TPA$TYPE
                U.147:  .WORD   5622                                    ;
    0000* 0011E ;TPA$TARGET
                U.148:  .WORD   <<U.103-U.148>-2>                       ;
          00120 GRPRO:  .BLKB   0
    7052  00120 ;TPA$TYPE
                U.149:  .WORD   28754                                   ;
00000000* 00122 ;TPA$ADDR
                U.150:  .LONG   <<PROT_VAL-U.150>-4>                    ;
00000100  00126 ;TPA$MASK
                U.151:  .LONG   256                                     ;
    0000* 0012A ;TPA$TARGET
                U.152:  .WORD   <<GRPRO-U.152>-2>                       ;
    7057  0012C ;TPA$TYPE
                U.153:  .WORD   28759                                   ;
00000000* 0012E ;TPA$ADDR
                U.154:  .LONG   <<PROT_VAL-U.154>-4>                    ;
00000200  00132 ;TPA$MASK
                U.155:  .LONG   512                                     ;
    0000* 00136 ;TPA$TARGET
                U.156:  .WORD   <<GRPRO-U.156>-2>                       ;
    7045  00138 ;TPA$TYPE
                U.157:  .WORD   28741                                   ;
00000000* 0013A ;TPA$ADDR
                U.158:  .LONG   <<PROT_VAL-U.158>-4>                    ;
00000400  0013E ;TPA$MASK
                U.159:  .LONG   1024                                    ;
    0000* 00142 ;TPA$TARGET
                U.160:  .WORD   <<GRPRO-U.160>-2>                       ;
    7044  00144 ;TPA$TYPE
                U.161:  .WORD   28740                                   ;
00000000* 00146 ;TPA$ADDR
                U.162:  .LONG   <<PROT_VAL-U.162>-4>                    ;
00000800  0014A ;TPA$MASK
                U.163:  .LONG   2048                                    ;
```

```
                        0000* 0014E ;TPA$TARGET
                              U.164:  .WORD   <<GRPRO-U.164>-2>                       ;
                        15F6  00150 ;TPA$TYPE
                              U.165:  .WORD   5622                                    ;
                        0000* 00152 ;TPA$TARGET
                              U.166:  .WORD   <<U.103-U.166>-2>                       ;
                              00154 ;WOPR
                              U.97:   .BLKB   0
                        003A  00154 ;TPA$TYPE
                              U.167:  .WORD   58                                      ;
                        003D  00156 ;TPA$TYPE
                              U.168:  .WORD   61                                      ;
                        15F6  00158 ;TPA$TYPE
                              U.169:  .WORD   5622                                    ;
                        0000* 0015A ;TPA$TARGET
                              U.170:  .WORD   <<U.103-U.170>-2>                       ;
                              0015C WOPR0:  .BLKB   0
                        7052  0015C ;TPA$TYPE
                              U.171:  .WORD   28754                                   ;
                    00000000* 0015E ;TPA$ADDR
                              U.172:  .LONG   <<PROT_VAL-U.172>-4>                    ;
                    00001000  00162 ;TPA$MASK
                              U.173:  .LONG   4096                                    ;
                        0000* 00166 ;TPA$TARGET
                              U.174:  .WORD   <<WOPR0-U.174>-2>                       ;
                        7057  00168 ;TPA$TYPE
                              U.175:  .WORD   28759                                   ;
                    00000000* 0016A ;TPA$ADDR
                              U.176:  .LONG   <<PROT_VAL-U.176>-4>                    ;
                    00002000  0016E ;TPA$MASK
                              U.177:  .LONG   8192                                    ;
                        0000* 00172 ;TPA$TARGET
                              U.178:  .WORD   <<WOPR0-U.178>-2>                       ;
                        7045  00174 ;TPA$TYPE
                              U.179:  .WORD   28741                                   ;
                    00000000* 00176 ;TPA$ADDR
                              U.180:  .LONG   <<PROT_VAL-U.180>-4>                    ;
                    00004000  0017A ;TPA$MASK
                              U.181:  .LONG   16384                                   ;
                        0000* 0017E ;TPA$TARGET
                              U.182:  .WORD   <<WOPR0-U.182>-2>                       ;
                        7044  00180 ;TPA$TYPE
                              U.183:  .WORD   28740                                   ;
                    00000000* 00182 ;TPA$ADDR
                              U.184:  .LONG   <<PROT_VAL-U.184>-4>                    ;
                    00008000  00186 ;TPA$MASK
                              U.185:  .LONG   32768                                   ;
                        0000* 0018A ;TPA$TARGET
                              U.186:  .WORD   <<WOPR0-U.186>-2>                       ;
                        15F6  0018C ;TPA$TYPE
                              U.187:  .WORD   5622                                    ;
                        0000* 0018E ;TPA$TARGET
                              U.188:  .WORD   <<U.103-U.188>-2>                       ;
                              00190 ;ENDPRO
                              U.103:  .BLKB   0
                        102C  00190 ;TPA$TYPE
                              U.189:  .WORD   4140                                    ;
```

```
                0000* 00192 ;TPA$TARGET
                      U.190:  .WORD   <<NEXTPRO-U.190>-2>                     ;
                15F7  00194 ;TPA$TYPE
                      U.191:  .WORD   5623                                    ;
                FFFF  00196 ;TPA$TARGET
                      U.192:  .WORD   -1                                      ;
                      00198 UIC_STB::
                            .BLKB   0
                45EC  00198 ;TPA$TYPE
                      U.194:  .WORD   17900                                   ;
            00000000* 0019A ;TPA$ADDR
                      U.195:  .LONG   <<OWNER_UIC-U.195>-4>                   ;
                15F7  0019E ;TPA$TYPE
                      U.196:  .WORD   5623                                    ;
                FFFF  001A0 ;TPA$TARGET
                      U.197:  .WORD   -1                                      ;
                      001A2         .BLKB   2
                      001A4 INDEX_STB::
                            .BLKB   0
                6100  001A4 ;TPA$TYPE
                      U.202:  .WORD   24832                                   ;
            00000000* 001A6 ;TPA$ADDR
                      U.203:  .LONG   <<INIT_OPTIONS-U.203>-4>                ;
            00100000  001AA ;TPA$MASK
                      U.204:  .LONG   1048576                                 ;
                6101  001AE ;TPA$TYPE
                      U.208:  .WORD   24833                                   ;
            00000000* 001B0 ;TPA$ADDR
                      U.209:  .LONG   <<INIT_OPTIONS-U.209>-4>                ;
            00200000  001B4 ;TPA$MASK
                      U.210:  .LONG   2097152                                 ;
                6102  001B8 ;TPA$TYPE
                      U.214:  .WORD   24834                                   ;
            00000000* 001BA ;TPA$ADDR
                      U.215:  .LONG   <<INIT_OPTIONS-U.215>-4>                ;
            00400000  001BE ;TPA$MASK
                      U.216:  .LONG   4194304                                 ;
                E5F3  001C2 ;TPA$TYPE
                      U.217:  .WORD   -6669                                   ;
            00000000* 001C4 ;TPA$ACTION
                      U.218:  .LONG   <<GET_INDEX_LBN-U.218>-4>               ;
            00000000* 001C8 ;TPA$ADDR
                      U.219:  .LONG   <<INIT_OPTIONS-U.219>-4>                ;
            00800000  001CC ;TPA$MASK
                      U.220:  .LONG   8388608                                 ;
                15F7  001D0 ;TPA$TYPE
                      U.222:  .WORD   5623                                    ;
                FFFF  001D2 ;TPA$TARGET
                      U.223:  .WORD   -1                                      ;
                      001D4 DATACHECK_STB::
                            .BLKB   0
                71F7  001D4 ;TPA$TYPE
                      U.225:  .WORD   29175                                   ;
            00000000* 001D6 ;TPA$ADDR
                      U.226:  .LONG   <<INIT_OPTIONS-U.226>-4>                ;
            00000100  001DA ;TPA$MASK
                      U.227:  .LONG   256                                     ;
```

```
                FFFF  001DE ;TPA$TARGET
                            U.228:  .WORD   -1                                          ;
                05F6  001E0 ;TPA$TYPE
                            U.229:  .WORD   1526                                        ;
                      001E2 CHECKOPT:
                            .BLKB   0
                6100  001E2 ;TPA$TYPE
                            U.233:  .WORD   24832                                       ;
            00000000* 001E4 ;TPA$ADDR
                            U.234:  .LONG   <<INIT_OPTIONS-U.234>-4>                    ;
            00000080  001E8 ;TPA$MASK
                            U.235:  .LONG   128                                         ;
                6501  001EC ;TPA$TYPE
                            U.239:  .WORD   25857                                       ;
            00000000* 001EE ;TPA$ADDR
                            U.240:  .LONG   <<INIT_OPTIONS-U.240>-4>                    ;
            00000100  001F2 ;TPA$MASK
                            U.241:  .LONG   256                                         ;
                102C  001F6 ;TPA$TYPE
                            U.243:  .WORD   4140                                        ;
                0000* 001F8 ;TPA$TARGET
                            U.244:  .WORD   <<CHECKOPT-U.244>-2>                        ;
                15F7  001FA ;TPA$TYPE
                            U.245:  .WORD   5623                                        ;
                FFFF  001FC ;TPA$TARGET
                            U.246:  .WORD   -1                                          ;
                      001FE         .BLKB   2
                      00200 BADBLOCKS_STB::
                                    .BLKB   0
                      00200 NEXTBLK:.BLKB   0
                85F3  00200 ;TPA$TYPE
                            U.248:  .WORD   -31245                                      ;
            00000000* 00202 ;TPA$ACTION
                            U.249:  .LONG   <<GET_BAD_LBN-U.249>-4>                     ;
                102E  00206 ;TPA$TYPE
                            U.250:  .WORD   4142                                        ;
                0000* 00208 ;TPA$TARGET
                            U.252:  .WORD   <<U.251-U.252>-2>                           ;
                05F6  0020A ;TPA$TYPE
                            U.253:  .WORD   1526                                        ;
                      0020C COLON:  .BLKB   0
                103A  0020C ;TPA$TYPE
                            U.254:  .WORD   4154                                        ;
                0000* 0020E ;TPA$TARGET
                            U.256:  .WORD   <<U.255-U.256>-2>                           ;
                05F6  00210 ;TPA$TYPE
                            U.257:  .WORD   1526                                        ;
                      00212 BLKEND: .BLKB   0
                102C  00212 ;TPA$TYPE
                            U.258:  .WORD   4140                                        ;
                0000* 00214 ;TPA$TARGET
                            U.259:  .WORD   <<NEXTBLK-U.259>-2>                         ;
                15F7  00216 ;TPA$TYPE
                            U.260:  .WORD   5623                                        ;
                FFFF  00218 ;TPA$TARGET
                            U.261:  .WORD   -1                                          ;
                      0021A ;BLKCNT
```

```
                        U.255:  .BLKB   0
    95F3  0021A ;TPA$TYPE
                        U.262:  .WORD   -27149                                  ;
00000000* 0021C ;TPA$ACTION
                        U.263:  .LONG   <<GET_BAD_COUNT-U.263>-4>               ;
    0000* 00220 ;TPA$TARGET
                        U.264:  .WORD   <<BLKEND-U.264>-2>                      ;
          00222 ;TRACK
                        U.251:  .BLKB   0
    85F3  00222 ;TPA$TYPE
                        U.265:  .WORD   -31245                                  ;
00000000* 00224 ;TPA$ACTION
                        U.266:  .LONG   <<GET_BAD_TRACK-U.266>-4>               ;
    042E  00228 ;TPA$TYPE
                        U.267:  .WORD   1070                                    ;
    95F3  0022A ;TPA$TYPE
                        U.268:  .WORD   -27149                                  ;
00000000* 0022C ;TPA$ACTION
                        U.269:  .LONG   <<GET_BAD_CYL-U.269>-4>                 ;
    0000* 00230 ;TPA$TARGET
                        U.270:  .WORD   <<COLON-U.270>-2>                       ;
          00232         .BLKB   2
          00234 LABEL_QUAL_STB::
                        .BLKB   0
          00234 LABEL_QUAL:
                        .BLKB   0
    7100  00234 ;TPA$TYPE
                        U.275:  .WORD   28928                                   ;
00000000* 00236 ;TPA$ADDR
                        U.276:  .LONG   <<<INIT_OPTIONS+4>-U.276>-4>            ;
00000040  0023A ;TPA$MASK
                        U.277:  .LONG   64                                      ;
    0000* 0023E ;TPA$TARGET
                        U.279:  .WORD   <<U.278-U.279>-2>                       ;
    7501  00240 ;TPA$TYPE
                        U.283:  .WORD   29953                                   ;
00000000* 00242 ;TPA$ADDR
                        U.284:  .LONG   <<<INIT_OPTIONS+4>-U.284>-4>            ;
00000080  00246 ;TPA$MASK
                        U.285:  .LONG   128                                     ;
    0000* 0024A ;TPA$TARGET
                        U.287:  .WORD   <<U.286-U.287>-2>                       ;
          0024C ;VOLUME
                        U.278:  .BLKB   0
    103A  0024C ;TPA$TYPE
                        U.289:  .WORD   4154                                    ;
    0000* 0024E ;TPA$TARGET
                        U.291:  .WORD   <<U.290-U.291>-2>                       ;
    143D  00250 ;TPA$TYPE
                        U.292:  .WORD   5181                                    ;
    0000* 00252 ;TPA$TARGET
                        U.293:  .WORD   <<U.290-U.293>-2>                       ;
          00254 ;VALUEVOLACC
                        U.290:  .BLKB   0
    95ED  00254 ;TPA$TYPE
                        U.294:  .WORD   -27155                                  ;
00000000* 00256 ;TPA$ACTION
```

```
                        U.295:  .LONG   <<GET_VOL_ACC-U.295>-4>                 ;
    0000* 0025A ;TPA$TARGET
                        U.297:  .WORD   <<U.296-U.297>-2>                       ;
          0025C ;OWNER
                        U.286:  .BLKB   0
    103A  0025C ;TPA$TYPE
                        U.298:  .WORD   4154                                    ;
    0000* 0025E ;TPA$TARGET
                        U.300:  .WORD   <<U.299-U.300>-2>                       ;
    143D  00260 ;TPA$TYPE
                        U.301:  .WORD   5181                                    ;
    0000* 00262 ;TPA$TARGET
                        U.302:  .WORD   <<U.299-U.302>-2>                       ;
          00264 ;VALUEOWNER
                        U.299:  .BLKB   0
    9022  00264 ;TPA$TYPE
                        U.303:  .WORD   -28638                                  ;
00000000* 00266 ;TPA$ACTION
                        U.304:  .LONG   <<MOVE_CHAR-U.304>-4>                   ;
    0000* 0026A ;TPA$TARGET
                        U.306:  .WORD   <<U.305-U.306>-2>                       ;
    902C  0026C ;TPA$TYPE
                        U.307:  .WORD   -28628                                  ;
00000000* 0026E ;TPA$ACTION
                        U.308:  .LONG   <<FORMAT_VOL_OWNER-U.308>-4>            ;
    0000* 00272 ;TPA$TARGET
                        U.309:  .WORD   <<LABEL_QUAL-U.309>-2>                  ;
    91F1  00274 ;TPA$TYPE
                        U.310:  .WORD   -28175                                  ;
00000000* 00276 ;TPA$ACTION
                        U.311:  .LONG   <<FORMAT_VOL_OWNER-U.311>-4>            ;
    0000* 0027A ;TPA$TARGET
                        U.312:  .WORD   <<U.296-U.312>-2>                       ;
    95F6  0027C ;TPA$TYPE
                        U.313:  .WORD   -27146                                  ;
00000000* 0027E ;TPA$ACTION
                        U.314:  .LONG   <<FORMAT_VOL_OWNER-U.314>-4>            ;
    0000* 00282 ;TPA$TARGET
                        U.315:  .WORD   <<U.296-U.315>-2>                       ;
          00284 ;ANSI_VOLO
                        U.305:  .BLKB   0
    9022  00284 ;TPA$TYPE
                        U.316:  .WORD   -28638                                  ;
00000000* 00286 ;TPA$ACTION
                        U.317:  .LONG   <<MOVE_CHAR-U.317>-4>                   ;
    0000* 0028A ;TPA$TARGET
                        U.318:  .WORD   <<U.299-U.318>-2>                       ;
    91ED  0028C ;TPA$TYPE
                        U.319:  .WORD   -28179                                  ;
00000000* 0028E ;TPA$ACTION
                        U.320:  .LONG   <<MOVE_CHAR-U.320>-4>                   ;
    0000* 00292 ;TPA$TARGET
                        U.321:  .WORD   <<U.305-U.321>-2>                       ;
    15F7  00294 ;TPA$TYPE
                        U.322:  .WORD   5623                                    ;
    FFFF  00296 ;TPA$TARGET
                        U.323:  .WORD   -1                                      ;
```

```
                00298 ;DONE
                      U.296:  .BLKB   0
102C            00298 ;TPA$TYPE
                      U.324:  .WORD   4140                                    ;
0000*           0029A ;TPA$TARGET
                      U.325:  .WORD   <<LABEL_QUAL-U.325>-2>                  ;
15F7            0029C ;TPA$TYPE
                      U.326:  .WORD   5623                                    ;
FFFF            0029E ;TPA$TARGET
                      U.327:  .WORD   -1                                      ;

                              .PSECT  _LIB$KEYO$,NOWRT,  SHR,  PIC,1

                00000 DENSITY_KTB::
                              .BLKB   0
                00000 ;TPA$KEYO
                      U.1:    .BLKB   0
0000*           00000 ;TPA$KEY
                      U.3:    .WORD   <U.2-U.1>                               ;
0000*           00002 ;TPA$KEY
                      U.9:    .WORD   <U.8-U.1>                               ;
0000*           00004 ;TPA$KEY
                      U.15:   .WORD   <U.14-U.1>                              ;
0000*           00006 ;TPA$KEY
                      U.21:   .WORD   <U.20-U.1>                              ;
0000*           00008 ;TPA$KEY
                      U.27:   .WORD   <U.26-U.1>                              ;
                0000A         .BLKB   2
                0000C STRUCTURE_KTB::
                              .BLKB   0
                0000C ;TPA$KEYO
                      U.35:   .BLKB   0
                0000C OVERRIDE_KTB::
                              .BLKB   0
                0000C ;TPA$KEYO
                      U.42:   .BLKB   0
0000*           0000C ;TPA$KEY
                      U.44:   .WORD   <U.43-U.42>                             ;
0000*           0000E ;TPA$KEY
                      U.50:   .WORD   <U.49-U.42>                             ;
0000*           00010 ;TPA$KEY
                      U.56:   .WORD   <U.55-U.42>                             ;
                00012         .BLKB   2
                00014 PROTECTION_KTB::
                              .BLKB   0
                00014 ;TPA$KEYO
                      U.66:   .BLKB   0
0000*           00014 ;TPA$KEY
                      U.68:   .WORD   <U.67-U.66>                             ;
0000*           00016 ;TPA$KEY
                      U.76:   .WORD   <U.75-U.66>                             ;
0000*           00018 ;TPA$KEY
                      U.84:   .WORD   <U.83-U.66>                             ;
0000*           C^01A ;TPA$KEY
                      U.92:   .WORD   <U.91-U.66>                             ;
                0001C UIC_KTB::
                              .BLKB   0
```

```
                        0001C ;TPA$KEY0
                              U.193:  .BLKB   0
                        0001C INDEX_KTB::
                                      .BLKB   0
                        0001C ;TPA$KEY0
                              U.198:  .BLKB   0
                  0000* 0001C ;TPA$KEY
                              U.200:  .WORD   <U.199-U.198>                           ;
                  0000* 0001E ;TPA$KEY
                              U.206:  .WORD   <U.205-U.198>                           ;
                  0000* 00020 ;TPA$KEY
                              U.212:  .WORD   <U.211-U.198>                           ;
                        00022         .BLKB   2
                        00024 DATACHECK_KTB::
                                      .BLKB   0
                        00024 ;TPA$KEY0
                              U.224:  .BLKB   0
                  0000* 00024 ;TPA$KEY
                              U.231:  .WORD   <U.230-U.224>                           ;
                  0000* 00026 ;TPA$KEY
                              U.237:  .WORD   <U.236-U.224>                           ;
                        00028 BADBLOCKS_KTB::
                                      .BLKB   0
                        00028 ;TPA$KEY0
                              U.247:  .BLKB   0
                        00028 LABEL_QUAL_KTB::
                                      .BLKB   0
                        00028 ;TPA$KEY0
                              U.271:  .BLKB   0
                  0000* 00028 ;TPA$KEY
                              U.273:  .WORD   <U.272-U.271>                           ;
                  0000* 0002A ;TPA$KEY
                              U.281:  .WORD   <U.280-U.271>                           ;

                                      .EXTRN  LIB$STOP
```

PSECT SUMMARY

Name	Bytes	Attributes
$GLOBAL$	936	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
OWN	52	NOVEC, WRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
$PLIT$	492	NOVEC,NOWRT, RD ,NOEXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
$CODE$	2080	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)
_LIB$KEY0$	44	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(1)
_LIB$STATE$	672	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(1)
_LIB$KEY1$	171	NOVEC,NOWRT, RD , EXE, SHR, LCL, REL, CON, PIC,ALIGN(1)

Library Statistics

File	Symbols Total	Symbols Loaded	Symbols Percent	Pages Mapped	Processing Time

```
;  _$255$DUA28:[SYSLIB]LIB.L32;1              18619        19         0       1000      00:01.8
;  _$255$DUA28:[SYSLIB]CLIMAC.L32;1              14         0         0          9      00:00.1
;  _$255$DUA28:[SYSLIB]TPAMAC.L32;1              42        28        66         14      00:00.2
```

```
;                                       COMMAND QUALIFIERS

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

; Size:         2080 code + 2367 data bytes
; Run Time:        01:15.5
; Elapsed Time:    02:33.9
; Lines/CPU Min:    1323
; Lexemes/CPU-Min: 73429
; Memory Used:  378 pages
; Compilation Complete
```