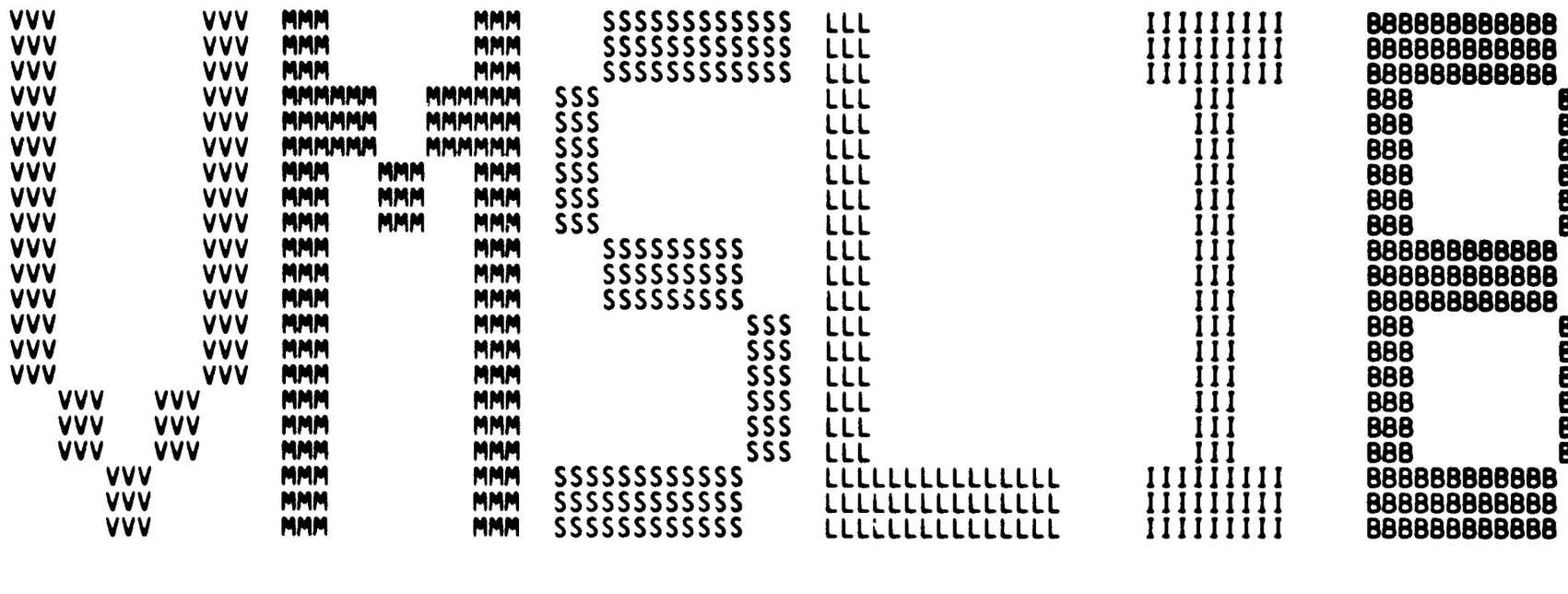
VVV VVV MMM MMM SSSSSSSSSSS LLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSSSS LLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSSSS LLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMMMMM MMMMMM SSS LLL III BBB BBB
VVV VVV MMMMMM MMMMMM SSS LLL III BBB BBB
VVV VVV MMMMMM MMMMMM SSS LLL III BBB BBB
VVV VVV MMM MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSSSSSSSS LLL III BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSS LLL III BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSS LLL III BBBBBBBBBBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSS LLL III BBB BBB
VVV VVV MMM MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBB
VVV VVV MMM MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBB
VVV MMM MMM SSSSSSSSSSS LLLLLLLLLLLLLLL IIIIIIIII BBBBBBBBBBB

FILEID**LIBACP

```
LL              IIIIII    BBBBBBBB      AAAAAA      CCCCCCCC  PPPPPPPP
LL              IIIIII    BBBBBBBB      AAAAAA      CCCCCCCC  PPPPPPPP
LL                II      BB      BB  AA      AA  CC          PP      PP
LL                II      BB      BB  AA      AA  CC          PP      PP
LL                II      BB      BB  AA      AA  CC          PP      PP
LL                II      BB      BB  AA      AA  CC          PP      PP
LL                II      BBBBBBBB    AA      AA  CC          PPPPPPPP
LL                II      BBBBBBBB    AA      AA  CC          PPPPPPPP
LL                II      BB      BB  AAAAAAAAAA  CC          PP
LL                II      BB      BB  AAAAAAAAAA  CC          PP
LL                II      BB      BB  AA      AA  CC          PP              ....
LL                II      BB      BB  AA      AA  CC          PP              ....
LLLLLLLLLL      IIIIII    BBBBBBBB    AA      AA    CCCCCCCC  PP              ....
LLLLLLLLLL      IIIIII    BBBBBBBB    AA      AA    CCCCCCCC  PP              ....

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 libacp (IDENT = 'V04-000') =
;     2          0002  1 BEGIN
;     3          0003  1
;     4          0004  1 !
;     5          0005  1 !*****************************************************************************
;     6          0006  1 !*                                                                           *
;     7          0007  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;     8          0008  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;     9          0009  1 !*  ALL RIGHTS RESERVED.                                                      *
;    10          0010  1 !*                                                                           *
;    11          0011  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    12          0012  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;    13          0013  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;    14          0014  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    15          0015  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    16          0016  1 !*  TRANSFERRED.                                                              *
;    17          0017  1 !*                                                                           *
;    18          0018  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    19          0019  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    20          0020  1 !*  CORPORATION.                                                              *
;    21          0021  1 !*                                                                           *
;    22          0022  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    23          0023  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    24          0024  1 !*                                                                           *
;    25          0025  1 !*                                                                           *
;    26          0026  1 !*****************************************************************************
;    27          0027  1
;    28          0028  1 !++
;    29          0029  1 ! FACILITY:  File system utility routines
;    30          0030  1 !
;    31          0031  1 ! ABSTRACT:
;    32          0032  1 !
;    33          0033  1 !       This module contains routines to manipulate the
;    34          0034  1 !       information in file headers.
;    35          0035  1 !
;    36          0036  1 ! ENVIRONMENT:
;    37          0037  1 !
;    38          0038  1 !       VAX/VMS operating system. unprivileged user mode,
;    39          0039  1 !
;    40          0040  1 ! AUTHOR:  Tim Halvorsen, Oct 1979
;    41          0041  1 !
;    42          0042  1 ! Modified by:
;    43          0043  1 !
;    44          0044  1 !       V03-003 TSK0001         Tamar Krichevsky        29-Jun-1984
;    45          0045  1 !               In LIB$CHECK_DIR use resultant string descriptor as the
;    46          0046  1 !               device name descriptor for the $ASSIGN system service,
;    47          0047  1 !               instead of the expanded name string descriptor.
;    48          0048  1 !
;    49          0049  1 !       V03-002 ACG0349         Andrew C. Goldstein,    5-Aug-1983  18:40
;    50          0050  1 !               Fix descriptor initialization in LIB$SET_ERASE
;    51          0051  1 !
;    52          0052  1 !       V03-001 ACG0331         Andrew C. Goldstein,    18-Apr-1983  17:28
;    53          0053  1 !               Convert LIB$SET_ERASE to set erase bit in file header
;    54          0054  1 !
;    55          0055  1 !       V02-014 MLJ0066         Martin L. Jack, 31-Dec-1981  9:51
;    56          0056  1 !               Split most routines out into separate modules.  Correct
;    57          0057  1 !               errors in LIB$SET_ERASE.
```

```
;    58     0058  1 !
;    59     0059  1 !       V02-013 SHZ0001         Stephen H. Zalewski,    11-Dec-1981  16:53
;    60     0060  1 !               Fixed LIB$FID_TO_NAME so that it does a $GETDVI for LOGVOLNAM.
;    61     0061  1 !               Also added code to insert a question mark into directory
;    62     0062  1 !               structure if the backlinks teminated other than at MFD.
;    63     0063  1 !
;    64     0064  1 !       V012    GRR2012         Greg Robert     16-Nov-1981
;    65     0065  1 !               Return SS$_NONLOCAL when node specified in create
;    66     0066  1 !               directory or set protection operations.
;    67     0067  1 !
;    68     0068  1 !       V011    TMH0011         Tim Halvorsen   20-Aug-1981
;    69     0069  1 !               Fix missing colon in resultant string from LIB$FID_TO_NAME.
;    70     0070  1 !
;    71     0071  1 !       V02-010 MLJ0028         Martin L. Jack, 8-Jul-1981  19:05
;    72     0072  1 !               Extend comparisons on FID$W_NUM to include FID$B_NMX.
;    73     0073  1 !               Clean up illegal up-level reference to NULLPARAMETER.
;    74     0074  1 !
;    75     0075  1 !       V009    GRR2009         Greg Robert     15-Jun-1981
;    76     0076  1 !               Utilized extended name block features to parse
;    77     0077  1 !               input name and to prevent calling ASSIGN system
;    78     0078  1 !               service with device name longer than 63 characters.
;    79     0079  1 !
;    80     0080  1 !       V008    TMH0008         Tim Halvorsen   12-Mar-1981
;    81     0081  1 !               Accept parameters to FID_TO_NAME as descriptors
;    82     0082  1 !               rather than vectors.
;    83     0083  1 !
;    84     0084  1 !       V007    TMH0007         Tim Halvorsen   27-Feb-1981
;    85     0085  1 !               In FID_TO_NAME, if RVN of backlink is zero,
;    86     0086  1 !               use RVN of file itself (RVN=0 is shorthand
;    87     0087  1 !               for "same volume").  Reference RTL routines
;    88     0088  1 !               with general addressing mode.
;    89     0089  1 !
;    90     0090  1 !       V02-006 ACG0184         Andrew C. Goldstein,    14-Jan-1981  11:01
;    91     0091  1 !               Add LIB$SET_ERASE, temporary implementation
;    92     0092  1 !
;    93     0093  1 !       V005    KRM0004         Karl Malik      14-Jan-1981
;    94     0094  1 !               Modified LIB$CHECK_DIR to recognize network
;    95     0095  1 !               directory filespecs.
;    96     0096  1 !
;    97     0097  1 !       V004    TMH0004         Tim Halvorsen   05-Jan-1981
;    98     0098  1 !               Fix LIB$FID_TO_NAME to work even though the backlinks
;    99     0099  1 !               may point to an unknown file.
;   100     0100  1 !
;   101     0101  1 !       003     TMH0003         Tim Halvorsen   17-Mar-1980
;   102     0102  1 !               Add LIB$FID_TO_NAME routine.
;   103     0103  1 !
;   104     0104  1 !       002     TMH0002         Tim Halvorsen   10-Mar-1980
;   105     0105  1 !               Drop delete access for all access modes when propagating
;   106     0106  1 !               protection from parent (because MFD has standard file
;   107     0107  1 !               protection on inited volume including delete access, but
;   108     0108  1 !               is protected from deletion by special check in ACP).
;   109     0109  1 !
;   110     0110  1 !       001     TMH0001         Tim Halvorsen   28-Jan-1980
;   111     0111  1 !               Support UIC format creation of directories.  Rearrange
;   112     0112  1 !               code so that illegal expanded name string won't leave
;   113     0113  1 !               the channel assigned.  Use protection of parent directory
;   114     0114  1 !               rather than process default protection on created directory.
```

```
;  115          0115  1 !--
;  116          0116  1
;  117          0117  1 !
;  118          0118  1 ! Include files
;  119          0119  1 !
;  120          0120  1
;  121          0121  1 LIBRARY 'SYS$LIBRARY:LIB.L32';          ! VMS system definitions
```

```
;  123          0122  1 !
;  124          0123  1 ! Table of contents
;  125          0124  1 !
;  126          0125  1
;  127          0126  1 FORWARD ROUTINE
;  128          0127  1         lib$check_dir,                  ! Check if directory file
;  129          0128  1         lib$set_erase,                  ! Mark file for erase-on-delete
;  130          0129  1         setup_fib;                      ! Common FIB initialization routine
;  131          0130  1
;  132          0131  1 !
;  133          0132  1 ! Define BBLOCK = BLOCK[,BYTE]
;  134          0133  1 !
;  135          0134  1
;  136          0135  1 STRUCTURE
;  137          0136  1         BBLOCK [O, P, S, E; N] =
;  138          0137  1                 [N]
;  139          0138  1                 (BBLOCK+O)<P,S,E>;
;  140          0139  1
;  141          0140  1 !
;  142          0141  1 ! Define various literal values
;  143          0142  1 !
;  144          0143  1
;  145          0144  1 LITERAL
;  146          0145  1     true        = 1,                    ! Boolean true
;  147          0146  1     false       = 0;                    ! Boolean false
;  148          0147  1
;  149          0148  1 !
;  150          0149  1 ! External routines
;  151          0150  1 !
;  152          0151  1
;  153          0152  1 EXTERNAL ROUTINE
;  154          0153  1     lib$get_vm: ADDRESSING_MODE(GENERAL),       ! Virtual memory allocation
;  155          0154  1     lib$free_vm: ADDRESSING_MODE(GENERAL);      ! Free virtual memory
;  156          0155  1
;  157          0156  1 !
;  158          0157  1 ! $FAB_DEV - macro to access FAB$L_DEV bits of FAB block.
;  159          0158  1 !
;  160          0159  1
;  161          0160  1 MACRO
;  162        M 0161  1     $fab_dev(dev_bit) =
;  163        M 0162  1         $BYTEOFFSET(fab$l_dev),
;  164          0163  1         $BITPOSITION(%NAME('dev$v_',dev_bit)),1,0%;
;  165          0164  1
;  166          0165  1 !
;  167          0166  1 ! DESCRIPTOR - define descriptor of static string
;  168          0167  1 !
;  169          0168  1
;  170          0169  1 MACRO
;  171        M 0170  1     descriptor(string) =
;  172          0171  1         UPLIT(%CHARCOUNT(string),UPLIT BYTE (string))%;
;  173          0172  1
;  174          0173  1 !
;  175          0174  1 ! Define macros to check status
;  176          0175  1 !
;  177          0176  1
;  178          0177  1 MACRO
;  179        M 0178  1     check_io =
```

```
;  180          M 0179  1          BEGIN
;  181          M 0180  1          IF .status                          ! If submitted ok,
;  182          M 0181  1          THEN
;  183          M 0182  1              status = .iosb [0];             ! then pick up I/O status
;  184          M 0183  1
;  185          M 0184  1          IF NOT .status                      ! If error detected,
;  186          M 0185  1          THEN
;  187          M 0186  1              BEGIN
;  188          M 0187  1              $DASSGN (CHAN = .channel);      ! Deassign channel
;  189          M 0188  1              RETURN .status;                 ! and report error
;  190          M 0189  1              END;
;  191            0190  1          END%,
;  192            0191  1
;  193          M 0192  1      perform (command) =
;  194          M 0193  1          BEGIN
;  195          M 0194  1          LOCAL status;
;  196          M 0195  1          status = command;
;  197          M 0196  1          IF NOT .status
;  198          M 0197  1          THEN
;  199          M 0198  1              RETURN .status;
;  200            0199  1          END%;
```

```
;  202     0200  1 GLOBAL ROUTINE lib$check_dir (fab_block) =
;  203     0201  1
;  204     0202  1 !----
;  205     0203  1 ! Functional description
;  206     0204  1 !
;  207     0205  1 !       This routine determines whether the file currently open
;  208     0206  1 !       by the specified FAB is a directory file or not.
;  209     0207  1 !
;  210     0208  1 ! Input parameters
;  211     0209  1 !
;  212     0210  1 !       fab_block       - FAB associated with the opened file.
;  213     0211  1 !
;  214     0212  1 !       The FAB is assumed to have an associated NAM block
;  215     0213  1 !       containing the FID of the file and a result name string.
;  216     0214  1 !
;  217     0215  1 ! Routine value
;  218     0216  1 !
;  219     0217  1 !       TRUE            - The file is a directory
;  220     0218  1 !       ss$_badirectory - The file is not a directory
;  221     0219  1 !       status          - Error was detected, assume not a directory
;  222     0220  1 !----
;  223     0221  1
;  224     0222  2 BEGIN
;  225     0223  2
;  226     0224  2 MAP
;  227     0225  2     fab_block: REF BBLOCK;              ! Address the input fab
;  228     0226  2
;  229     0227  2 BIND
;  230     0228  2     nam_block = .fab_block [fab$l_nam]: BBLOCK;
;  231     0229  2
;  232     0230  2 LOCAL
;  233     0231  2     fib:        BBLOCK[fib$c_accdata],  ! File Identification Block
;  234     0232  2     fib_desc:   BBLOCK[8],              ! FIB descriptor
;  235     0233  2     atr:        BLOCKVECTOR[4,8,BYTE],  ! Attribute control block
;  236     0234  2     filatr:     BBLOCK[atr$s_recattr]   ! File attributes
;  237     0235  2                 VOLATILE,
;  238     0236  2     header:     BBLOCK[atr$s_header]    ! File header block
;  239     0237  2                 VOLATILE,
;  240     0238  2     dev_desc:   BBLOCK[8],              ! Device descriptor for ASSIGN
;  241     0239  2     channel:    WORD,                   ! Channel to device
;  242     0240  2     iosb:       VECTOR[4,WORD],         ! I/O status block
;  243     0241  2     status;                             ! Holds RMS status codes
;  244     0242  2
;  245     0243  2 !
;  246     0244  2 ! Check the file type and version.  A valid directory must
;  247     0245  2 ! be named .DIR;1 or else it is invalid.
;  248     0246  2 !
;  249     0247  2
;  250     0248  2 IF CH$FIND_SUB(.nam_block [nam$b_rsl], .nam_block [nam$l_rsa],
;  251     0249  2         6,UPLIT('.DIR;1')) EQL 0        ! If not .DIR
;  252     0250  2 THEN
;  253     0251  2     RETURN ss$_badirectory;             ! then not a valid directory
;  254     0252  2 !
;  255     0253  2 ! If this is a network directory filespec then do not attempt to
;  256     0254  2 ! assign a channel for QIO operations - just return the appropriate
;  257     0255  2 ! return value.
;  258     0256  2 !
```

```
;  259          0257  3 IF (.fab_block[$fab_dev(net)])          ! If this is a network operation
;  260          0258  2 THEN
;  261          0259  3     BEGIN
;  262          0260  3     IF (.fab_block[fab$b_rat]) EQL fab$m_blk !and there is no carriage control
;  263          0261  3     OR (.fab_block[fab$b_rat]) EQL 0 !
;  264          0262  3     THEN
;  265          0263  3         RETURN true                       ! It's a valid network directory
;  266          0264  3     ELSE
;  267          0265  3         RETURN ss$_badirectory;           ! It's not a valid network directory
;  268          0266  2     END;
;  269          0267  2 !
;  270          0268  2 ! Assign a channel to the device for QIO operations.  If an error
;  271          0269  2 ! occurs, then exit without success.
;  272          0270  2 !
;  273          0271  2
;  274          0272  2 dev_desc [dsc$w_length] = .nam_block [nam$b_rss];
;  275          0273  2 dev_desc [dsc$a_pointer] = .nam_block [nam$l_rsa];
;  276          0274  2
;  277        P 0275  2 perform($ASSIGN( DEVNAM = dev_desc,       ! Assign channel to device
;  278          0276  2                  CHAN = channel));
;  279          0277  2
;  280          0278  2 !
;  281          0279  2 ! Call the ACP to read the file header attributes with a single QIO.
;  282          0280  2 !
;  283          0281  2
;  284          0282  2 atr [0,atr$w_type] = atr$c_recattr;       ! Request file attributes
;  285          0283  2 atr [0,atr$w_size] = atr$s_recattr;
;  286          0284  2 atr [0,atr$l_addr] = filatr;
;  287          0285  2 atr [1,atr$w_type] = atr$c_header;        ! Request file header block
;  288          0286  2 atr [1,atr$w_size] = atr$s_header;
;  289          0287  2 atr [1,atr$l_addr] = header;
;  290          0288  2 atr [2,0,0,32,0] = 0;                    ! Trailing zero longword
;  291          0289  2
;  292          0290  2 fib_desc [dsc$w_length] = fib$c_accdata;
;  293          0291  2 fib_desc [dsc$a_pointer] = fib;
;  294          0292  2
;  295          0293  2 fib [fib$l_acctl] = 0;                    ! Allow readers and writers
;  296          0294  2 fib [fib$w_fid_num] = .nam_block [nam$w_fid_num];
;  297          0295  2 fib [fib$w_fid_seq] = .nam_block [nam$w_fid_seq];
;  298          0296  2 fib [fib$w_fid_rvn] = .nam_block [nam$w_fid_rvn];
;  299          0297  2
;  300        P 0298  2 status = $QIOW( CHAN = .channel,          ! Open and read file header
;  301        P 0299  2                 FUNC = IO$_ACCESS,
;  302        P 0300  2                 IOSB = iosb,              ! Address of I/O status block
;  303        P 0301  2                 P1 = fib_desc,            ! Descriptor of FIB block
;  304          0302  2                 P5 = atr);                ! Address of attribute block
;  305          0303  2
;  306          0304  2 check_io;                                 ! Check I/O status
;  307          0305  2
;  308          0306  2 !
;  309          0307  2 ! Deassign the channel used to access the device
;  310          0308  2 !
;  311          0309  2
;  312          0310  2 perform($DASSGN( CHAN = .channel));       ! Deassign the channel
;  313          0311  2
;  314          0312  2 !
;  315          0313  2 ! Check the file characteristics to determine if this file is
```

```
;  316          0314  2 ! really a directory file.
;  317          0315  2 !
;  318          0316  2
;  319          0317  2 IF .header [fh2$b_struclev] EQL 2       ! If ODS-2 format,
;  320          0318  2 THEN
;  321          0319  3     BEGIN
;  322          0320  3     IF NOT .header [fh2$v_directory]    ! DIRECTORY bit must be on
;  323          0321  3     THEN
;  324          0322  3         RETURN ss$_badirectory;         ! If not, exit not a directory
;  325          0323  3     END
;  326          0324  2 ELSE
;  327          0325  2 IF .header [fh2$b_struclev] EQL 1       ! If ODS-1 format,
;  328          0326  2 THEN
;  329          0327  3     BEGIN
;  330          0328  3     IF .filatr [fat$b_rtype] NEQ fat$c_fixed    ! Must be fixed records
;  331          0329  3         OR .filatr [fat$b_rattrib] NEQ 0        ! & no carriage control
;  332          0330  3     THEN
;  333          0331  3         RETURN ss$_badirectory;         ! If not, exit not a directory
;  334          0332  3     END
;  335          0333  2 ELSE
;  336          0334  2     RETURN ss$_badirectory;             ! If not ODS-1 or 2, bad directory
;  337          0335  2
;  338          0336  2 RETURN true;                            ! Return file is a directory file
;  339          0337  2
;  340          0338  1 END;


                                                                .TITLE  LIBACP
                                                                .IDENT  \V04-000\

                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                    00  00  31  3B  52  49  44  2E  00000 P.AAA:  .ASCII  \.DIR;1\<0><0>                          ;

                                                                .EXTRN  LIB$GET_VM, LIB$FREE_VM
                                                                .EXTRN  SYS$ASSIGN, SYS$QIOW
                                                                .EXTRN  SYS$DASSGN

                                                                .PSECT  $CODE$,NOWRT,2

                                                  007C 00000    .ENTRY  LIB$CHECK_DIR, Save R2,R3,R4,R5,R6          ; 0200
                                  56 00000000G    00   9E 00002 MOVAB   SYS$DASSGN, R6                            ;
                                  5E      FD98    CE   9E 00009 MOVAB   -616(SP), SP                              ;
                                  54        04    AC   D0 0000E MOVL    FAB_BLOCK, R4                             ; 0228
                                  55        28    A4   D0 00012 MOVL    40(R4), R5                                ;
                                  50        03    A5   9A 00016 MOVZBL  3(R5), R0                                 ; 0248
04  B5                  50  0000'  CF              06   39 0001A MATCHC  #6, P.AAA, R0, @4(R5)                    ; 0249
                                                    03   13 00022 BEQL    1$                                        ;
                                  53              06   D0 00024 MOVL    #6, R3                                    ;
                                  53              06   C2 00027 1$:    SUBL2   #6, R3                            ;
                                                    03   12 0002A BNEQ    3$                                        ;
                                                 00C0   31 0002C 2$:    BRW     10$                               ;
                      0E    41    A4                05   E1 0002F 3$:    BBC     #5, 65(R4), 5$                    ; 0257
                                  08        1E    A4   91 00034 CMPB    30(R4), #8                                ; 0260
                                                    05   13 00038 BEQL    4$                                        ;
                                            1E    A4   95 0003A TSTB    30(R4)                                    ; 0261
                                                  ED   12 0003D BNEQ    2$                                        ;
```

```
                                00B3  31 0003F 4$:     BRW     11$                                     ; 0265
            0C    AE       02   A5  9B 00042 5$:     MOVZBW  2(R5), DEV_DESC                         ; 0272
            10    AE       04   A5  D0 00047         MOVL    4(R5), DEV_DESC+4                       ; 0273
                                7E  7C 0004C         CLRQ    -(SP)                                   ; 0276
                             08 AE  9F 0004E         PUSHAB  CHANNEL
                             18 AE  9F 00051         PUSHAB  DEV_DESC
00000000G                    00     04 FB 00054      CALLS   #4, SYS$ASSIGN
                             70     50 E9 0005B      BLBC    STATUS, 8$
            CC    AD 00040020   8F  D0 0005E         MOVL    #262176, ATR                            ; 0283
            D0    AD       AC   AD  9E 00066         MOVAB   FILATR, ATR+4                           ; 0284
            D4    AD 000A0200   8F  D0 0006B         MOVL    #655872, ATR+8                          ; 0286
            D8    AD       14   AE  9E 00073         MOVAB   HEADER, ATR+12                          ; 0287
                           DC AD  D4 00078         CLRL    ATR+16                                  ; 0288
            EC    AD           0A  B0 0007B         MOVW    #10, FIB_DESC                           ; 0290
            F0    AD       F4   AD  9E 0007F         MOVAB   FIB, FIB_DESC+4                         ; 0291
                           F4 AD  D4 00084         CLRL    FIB                                     ; 0293
            F8    AD       24   A5  D0 00087         MOVL    36(R5), FIB+4                           ; 0294
            FC    AD       28   A5  B0 0008C         MOVW    40(R5), FIB+8                           ; 0296
                                7E  D4 00091         CLRL    -(SP)                                   ; 0302
                           CC AD  9F 00093         PUSHAB  ATR
                                7E  7C 00096         CLRQ    -(SP)
                                7E  D4 00098         CLRL    -(SP)
                           EC AD  9F 0009A         PUSHAB  FIB_DESC
                                7E  7C 0009D         CLRQ    -(SP)
                           24 AE  9F 0009F         PUSHAB  IOSB
                                32  DD 000A2         PUSHL   #50
                        7E 28 AE  3C 000A4         MOVZWL  CHANNEL, -(SP)
                                7E  D4 000A8         CLRL    -(SP)
00000000G                    00     0C FB 000AA      CALLS   #12, SYS$QIOW
                             52     50 D0 000B1      MOVL    R0, STATUS
                             07     52 E9 000B4      BLBC    STATUS, 6$
                             52  04 AE  3C 000B7      MOVZWL  IOSB, STATUS
                             0A     52 E8 000BB      BLBS    STATUS, 7$
                             7E     6E 3C 000BE 6$:  MOVZWL  CHANNEL, -(SP)
                             66     01 FB 000C1      CALLS   #1, SYS$DASSGN
                             50     52 D0 000C4      MOVL    STATUS, R0
                                    04 000C7         RET
                             7E     6E 3C 000C8 7$:  MOVZWL  CHANNEL, -(SP)                          ; 0310
                             66     01 FB 000CB      CALLS   #1, SYS$DASSGN
                             27     50 E9 000CE 8$:  BLBC    STATUS, 12$
                             02  1B AE  91 000D1         CMPB    HEADER+7, #2                    ; 0317
                                07  12 000D5         BNEQ    9$
                     19  49 AE        05 E0 000D7         BBS     #5, HEADER+53, 11$              ; 0320
                                11  11 000DC         BRB     10$                                     ; 0322
                             01  1B AE  91 000DE 9$:  CMPB    HEADER+7, #1                    ; 0325
                                0B  12 000E2         BNEQ    10$
                             01  AC AD  91 000E4         CMPB    FILATR, #1                      ; 0328
                                05  12 000E8         BNEQ    10$
                                AD  AD  95 000EA         TSTB    FILATR+1                        ; 0329
                                06  13 000ED         BEQL    11$
                             50 0828 8F  3C 000EF 10$:  MOVZWL  #2088, R0                       ; 0334
                                    04 000F4         RET
                             50     01  D0 000F5 11$:  MOVL    #1, R0                          ; 0336
                                    04 000F8 12$:  RET                                             ; 0338

; Routine Size:  249 bytes,    Routine Base:  $CODE$ + 0000
```

```
;  342     0339  1 GLOBAL ROUTINE lib$set_erase (name_desc) =
;  343     0340  1
;  344     0341  1 !---
;  345     0342  1 !
;  346     0343  1 ! Functional description
;  347     0344  1 !
;  348     0345  1 !       This routine sets the erase-on-delete bit in a file.
;  349     0346  1 !
;  350     0347  1 ! Inputs:
;  351     0348  1 !
;  352     0349  1 !       name_desc = Address of descriptor of directory file name
;  353     0350  1 !
;  354     0351  1 ! Outputs:
;  355     0352  1 !
;  356     0353  1 !       success or failure status
;  357     0354  1 !---
;  358     0355  1
;  359     0356  2 BEGIN
;  360     0357  2
;  361     0358  2 MAP
;  362     0359  2     name_desc:  REF BBLOCK;             ! Address of name descriptor
;  363     0360  2
;  364     0361  2 LOCAL
;  365     0362  2     fib:        BBLOCK[fib$c_extdata],  ! File Identification Block
;  366     0363  2     fib_desc:   BBLOCK[8],              ! FIB descriptor
;  367     0364  2     channel:    WORD,                   ! Channel to device
;  368     0365  2     iosb:       VECTOR[4,WORD],         ! I/O status block
;  369     0366  2     status,                             ! Holds RMS status codes
;  370     0367  2     atr_list:   BBLOCK[12],             ! Attribute control list
;  371     0368  2     file_char:  BBLOCK[4];              ! File charactreistics longword
;  372     0369  2
;  373     0370  2 !
;  374     0371  2 ! Call the common setup routine to get the file ID of the file and
;  375     0372  2 ! set up the FIB. Set up an attribute list to read the file characteristics.
;  376     0373  2 !
;  377     0374  2
;  378     0375  2 perform (setup_fib (.name_desc, fib, channel));
;  379     0376  2
;  380     0377  2 fib_desc [dsc$w_length] = fib$c_extdata; ! Create FIB descriptor
;  381     0378  2 fib_desc [dsc$a_pointer] = fib;
;  382     0379  2 fib_desc [dsc$b_dtype] = 0;
;  383     0380  2 fib_desc [dsc$b_class] = 0;
;  384     0381  2
;  385     0382  2 fib [fib$l_acctl] = fib$m_write;
;  386     0383  2
;  387     0364  2 atr_list [atr$w_size] = atr$s_uchar;
;  388     0385  2 atr_list [atr$w_type] = atr$c_uchar;
;  389     0386  2 atr_list [atr$l_addr] = file_char;
;  390     0387  2 atr_list [8, 0, 32, 0] = 0;
;  391     0388  2
;  392   P 0389  2 status = $QIOW (CHAN = .channel,          ! open the file for update
;  393   P 0390  2         FUNC = IO$_ACCESS OR IO$M_ACCESS,
;  394   P 0391  2         IOSB = iosb,
;  395   P 0392  2         P1 = fib_desc,
;  396   P 0393  2         P5 = atr_list);
;  397     0394  2
;  398     0395  2 check_io;                               ! Check I/O status
```

```
;  399          0396  2
;  400          0397  2 ! Set the erase bit in the file characteristics and deaccess the file,
;  401          0398  2 ! writing the characterictics longword back.
;  402          0399  2 !
;  403          0400  2
;  404          0401  2 file_char[fch$v_erase] = 1;
;  405          0402  2
;  406        P 0403  2 status = $QIOW (CHAN = .channel,
;  407        P 0404  2                 FUNC = IO$_DEACCESS,
;  408        P 0405  2                 IOSB = iosb,
;  409          0406  2                 P5 = atr_list);
;  410          0407  2
;  411          0408  2 check_io;                               ! Check I/O status
;  412          0409  2
;  413          0410  2 !
;  414          0411  2 !       Deassign the channel
;  415          0412  2 !
;  416          0413  2
;  417          0414  2 perform ($DASSGN (CHAN = .channel));    ! Deassign the channel
;  418          0415  2
;  419          0416  2 RETURN .status;                         ! Return successful
;  420          0417  2
;  421          0418  1 END;


                                003C 00000          .ENTRY  LIB$SET_ERASE, Save R2,R3,R4,R5          ; 0339
                     55 00000000G   00   9E 00002    MOVAB   SYS$DASSGN, R5
                     54 00000000G   00   9E 00009    MOVAB   SYS$QIOW, R4
                     5E        BC   AE   9E 00010    MOVAB   -68(SP), SP
                                    5E   DD 00014    PUSHL   SP                                       ; 0375
                               28   AE   9F 00016    PUSHAB  FIB
                               04   AC   DD 00019    PUSHL   NAME_DESC
                0000V          CF        03   FB 0001C    CALLS   #3, SETUP_FIB
                               01        50   E8 00021    BLBS    STATUS, 1$
                                         04   00024    RET
                20   AE        24   AE   9E 00025 1$:    MOVAB   FIB, FIB_DESC+4                  ; 0378
                1C   AE             20   D0 0002A    MOVL    #32, FIB_DESC                        ; 0377
                24   AE      0100   8F   3C 0002E    MOVZWL  #256, FIB                            ; 0382
                08   AE  00030004   8F   D0 00034    MOVL    #196612, ATR_LIST                    ; 0384
                0C   AE        04   AE   9E 0003C    MOVAB   FILE_CHAR, ATR_LIST+4                ; 0386
                               10   AE   D4 00041    CLRL    ATR_LIST+8                           ; 0387
                                    7E   D4 00044    CLRL    -(SP)                                ; 0393
                               0C   AE   9F 00046    PUSHAB  ATR_LIST
                                    7E   7C 00049    CLRQ    -(SP)
                                    7E   D4 0004B    CLRL    -(SP)
                               30   AE   9F 0004D    PUSHAB  FIB_DESC
                                    7E   7C 00050    CLRQ    -(SP)
                               34   AE   9F 00052    PUSHAB  IOSB
                     7E        72   8F   9A 00055    MOVZBL  #114, -(SP)
                     53        28   AE   3C 00059    MOVZWL  CHANNEL, R3
                                    53   DD 0005D    PUSHL   R3
                                    7E   D4 0005F    CLRL    -(SP)
                     64             0C   FB 00061    CALLS   #12, SYS$QIOW
                     52             50   D0 00064    MOVL    R0, STATUS
```

```
                2F           52  E9 00067          BLBC    STATUS, 2$                          ;
                52       14  AE  3C 0006A          MOVZWL  IOSB, STATUS                        ;
                28           52  E9 0006E          BLBC    STATUS, 2$                          ;
             06 AE           02  88 00071          BISB2   #2, FILE_CHAR+2                     ; 0401
                             7E  D4 00075          CLRL    -(SP)                               ; 0406
                         0C  AE  9F 00077          PUSHAB  ATR_LIST                            ;
                             7E  7C 0007A          CLRQ    -(SP)                               ;
                             7E  7C 0007C          CLRQ    -(SP)                               ;
                             7E  7C 0007E          CLRQ    -(SP)                               ;
                         34  AE  9F 00080          PUSHAB  IOSB                                ;
                             34  DD 00083          PUSHL   #52                                 ;
                             53  DD 00085          PUSHL   R3                                  ;
                             7E  D4 00087          CLRL    -(SP)                               ;
                64           0C  FB 00089          CALLS   #12, SYS$QIOW                       ;
                52           50  D0 0008C          MOVL    R0, STATUS                          ;
                07           52  E9 0008F          BLBC    STATUS, 2$                          ;
                52       14  AE  3C 00092          MOVZWL  IOSB, STATUS                        ;
                07           52  E8 00096          BLBS    STATUS, 3$                          ;
                             53  DD 00099 2$:       PUSHL   R3                                  ;
                65           01  FB 0009B          CALLS   #1, SYS$DASSGN                      ;
                             08  11 0009E          BRB     4$                                  ;
                             53  DD 000A0 3$:       PUSHL   R3                                  ; 0414
                65           01  FB 000A2          CALLS   #1, SYS$DASSGN                      ;
                03           50  E9 000A5          BLBC    STATUS, 5$                          ;
                50           52  D0 000A8 4$:       MOVL    STATUS, R0                          ; 0416
                                 04 000AB 5$:       RET                                         ; 0418

; Routine Size:  172 bytes,    Routine Base:  $CODE$ + 00F9
```



```
$$.
$$.
$$T
ASS
ASS
FAB
FAB
FAB
FAB
FAB
FAB
FAB
FAB
FAB
FAB
LIB
LIB
LIB
LIB
NAM
NAM
NAM
NAM
NAM
NAM
NAM
NAM
NAM
NAM
NAM
NAM
RMS
RMS
SS$
SYS
SYS
SYS
SYS


PSE
---

$AB
_LI


Pha
---
Ini
Com
Pas
Sym
```

```
;  423          0419  1 ROUTINE setup_fib (name_desc, fib, channel) =
;  424          0420  1
;  425          0421  1 !---
;  426          0422  1 !
;  427          0423  1 ! Functional description
;  428          0424  1 !
;  429          0425  1 !       This routine parses the specified file name, fills in the fib
;  430          0426  1 !       with the file ID, and assigns a channel to the device.
;  431          0427  1 !
;  432          0428  1 ! Inputs:
;  433          0429  1 !
;  434          0430  1 !       name_desc = Address of descriptor of directory file name
;  435          0431  1 !
;  436          0432  1 ! Outputs:
;  437          0433  1 !
;  438          0434  1 !       fib = address of fib to be filled in
;  439          0435  1 !       channel = address of word to return channel number
;  440          0436  1 !
;  441          0437  1 ! Value:
;  442          0438  1 !
;  443          0439  1 !       success or failure status code
;  444          0440  1 !
;  445          0441  1 !---
;  446          0442  1
;  447          0443  2 BEGIN
;  448          0444  2
;  449          0445  2 MAP
;  450          0446  2     name_desc:  REF BBLOCK,             ! Address of name descriptor
;  451          0447  2     fib:        REF BBLOCK;             ! File Identification Block
;  452          0448  2
;  453          0449  2 LOCAL
;  454          0450  2     fab:        BBLOCK [fab$c_bln],     ! FAB to open directory file
;  455          0451  2     nam:        BBLOCK [nam$c_bln],     ! NAM to obtain DID, etc.
;  456          0452  2     expbuf:     VECTOR [nam$c_maxrss,BYTE],
;  457          0453  2     desc:       VECTOR [2],             ! Descriptor
;  458          0454  2     status;                             ! Holds RMS status codes
;  459          0455  2
;  460          0456  2 !
;  461          0457  2 !       Parse the file specification with RMS to obtain the
;  462          0458  2 !       expanded name string.  RMS should return DNF but all
;  463          0459  2 !       that is needed is the expanded string.
;  464          0460  2 !
;  465          0461  2
;  466        P 0462  2 $FAB_INIT (FAB = fab,                     ! Initialize FAB block
;  467        P 0463  2           NAM = nam,
;  468        P 0464  2           FNA = .name_desc [dsc$a_pointer],
;  469          0465  2           FNS = .name_desc [dsc$w_length]);
;  470          0466  2
;  471        P 0467  2 $NAM_INIT (NAM = nam,                     ! Initialize NAM block
;  472        P 0468  2           ESA = expbuf,
;  473          0469  2           ESS = nam$c_maxrss);
;  474          0470  2
;  475          0471  2 status = $PARSE (FAB = fab);             ! Parse the input string
;  476          0472  2
;  477          0473  2 IF NOT .status                           ! If an unexpected error,
;  478          0474  2 THEN
;  479          0475  2     RETURN .status;                     ! exit with status
```

```
;   480          0476  2
;   481          0477  2 !
;   482          0478  2 !       If this is a network operation then return a "non-local
;   483          0479  2 !       device" error to the caller.
;   484          0480  2 !
;   485          0481  2
;   486          0482  2 IF .nam [nam$b_node] NEQ 0              ! If a node was specified
;   487          0483  2 THEN RETURN ss$_nonlocal;               ! then exit with error
;   488          0484  2
;   489          0485  2 IF .nam [nam$v_wildcard]               ! If wildcards specified,
;   490          0486  2 THEN
;   491          0487  2     RETURN ss$_nosuchfile;             ! then return no such file
;   492          0488  2
;   493          0489  2 status = $SEARCH (FAB = fab);          ! Get FID and DID fields
;   494          0490  2
;   495          0491  2 IF NOT .status                         ! If not found or error,
;   496          0492  2 THEN
;   497          0493  2     RETURN .status;                    ! then exit with status
;   498          0494  2
;   499          0495  2 !
;   500          0496  2 !       Assign a channel to the device ACP
;   501          0497  2 !
;   502          0498  2
;   503          0499  2 desc [0] = .nam [nam$b_dev];           ! Fetch the device name size
;   504          0500  2 desc [1] = .nam [nam$l_dev];           ! and location from the expanded name
;   505          0501  2
;   506        P 0502  2 perform ($ASSIGN (DEVNAM = desc,       ! Assign channel to ACP
;   507          0503  2                   CHAN = .channel));
;   508          0504  2
;   509          0505  2 !
;   510          0506  2 !       Setup parameters to be sent to the ACP
;   511          0507  2 !
;   512          0508  2
;   513          0509  2 CH$FILL(0,fib$c_extdata,.fib);         ! Zero the FIB first
;   514          0510  2
;   515          0511  2 fib [fib$w_fid_num] = .nam [nam$w_fid_num]; ! Copy FID
;   516          0512  2 fib [fib$w_fid_seq] = .nam [nam$w_fid_seq];
;   517          0513  2 fib [fib$w_fid_rvn] = .nam [nam$w_fid_rvn];
;   518          0514  2
;   519          0515  2 RETURN true;                           ! Return successful
;   520          0516  2
;   521          0517  1 END;


                                                                .EXTRN  SYS$PARSE, SYS$SEARCH

                                                    007C 00000 SETUP_FIB:
                                                                .WORD   Save R2,R3,R4,R5,R6                             ; 0419
                                    5E      FE48    CE  9E 00002        MOVAB   -440(SP), SP                            ;
    0050    8F              00      6E              00  2C 00007        MOVC5   #0, (SP), #0, #80, $RMS_PTR             ; 0465
                                            B0      AD     0000E        
                                B0  AD      5003    8F  B0 00010        MOVW    #20483, $RMS_PTR                        ;
                                C6  AD              02  90 00016        MOVB    #2, $RMS_PTR+22                         ;
                                CF  AD              02  90 0001A        MOVB    #2, $RMS_PTR+31                         ;
                                D8  AD      FF50    CD  9E 0001E        MOVAB   NAM, $RMS_PTR+40                        ;
                                    50      04      AC  D0 00024        MOVL    NAME_DESC, R0                           ;
```

```
                                DC   AD      04   A0   D0 00028          MOVL    4(R0), $RMS_PTR+44                          :
                                E4   AD           60   90 0002D          MOVB    (R0), $RMS_PTR+52                           :
    0060    8F          00           6E           00   2C 00031          MOVC5   #0, (SP), #0, #96, $RMS_PTR                 : 0469
                                                 FF50 CD    00038                                                           :
                            FF50 CD      6002    8F   B0 0003B          MOVW    #24578, $RMS_PTR                            :
                            FF5A CD              01   8E 00042          MNEGB   #1, $RMS_PTR+10                             :
                            FF5C CD        08    AE   9E 00047          MOVAB   EXPBUF, $RMS_PTR+12                         :
                                           B0    AD   9F 0004D          PUSHAB  FAB                                         : 0471
                    00000000G   00               01   FB 00050          CALLS   #1, SYS$PARSE                               :
                                56               50   E9 00057          BLBC    STATUS, 3$                                  : 0473
                                           88    AD   95 0005A          TSTB    NAM+56                                      : 0482
                                                 06   13 0005D          BEQL    1$                                          :
                                50       08F0    8F   3C 0005F          MOVZWL  #2288, R0                                   : 0483
                                                 04   00064            RET                                                 :
                                06         85    AD   E9 00065 1$:      BLBC    NAM+53, 2$                                  : 0485
                                50       0910    8F   3C 00069          MOVZWL  #2320, R0                                   : 0487
                                                 04   0006E            RET                                                 :
                                           B0    AD   9F 0006F 2$:      PUSHAB  FAB                                         : 0489
                    00000000G   00               01   FB 00072          CALLS   #1, SYS$SEARCH                              :
                                34               50   E9 00079          BLBC    STATUS, 3$                                  : 0491
                                6E         89    AD   9A 0007C          MOVZBL  NAM+57, DESC                                : 0499
                            04  AE         94    AD   D0 00080          MOVL    NAM+68, DESC+4                              : 0500
                                                 7E   7C 00085          CLRQ    -(SP)                                       : 0503
                                           0C    AC   DD 00087          PUSHL   CHANNEL                                     :
                                           0C    AE   9F 0008A          PUSHAB  DESC                                        :
                    00000000G   00               04   FB 0008D          CALLS   #4, SYS$ASSIGN                              :
                                19               50   E9 00094          BLBC    STATUS, 3$                                  :
                                56         08    AC   D0 00097          MOVL    FIB, R6                                     : 0509
        20          00           6E               00   2C 0009B          MOVC5   #0, (SP), #0, #32, (R6)                     :
                                                 66     000A0                                                             :
                            04  A6       FF74    CD   D0 000A1          MOVL    NAM+36, 4(R6)                               : 0511
                            08  A6       FF78    CD   B0 000A7          MOVW    NAM+40, 8(R6)                               : 0513
                                50               01   D0 000AD          MOVL    #1, R0                                      : 0515
                                                 04   000B0 3$:         RET                                                 : 0517

; Routine Size:  177 bytes,    Routine Base:  $CODE$ + 01A5
```

```
;  523          0518  1 END
;  524          0519  0 ELUDOM


;                                   PSECT SUMMARY

;       Name                    Bytes                           Attributes

;  $PLIT$                          8  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                        598  NOVEC,NOWRT,  RD ,  EXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)


;                                 Library Statistics

;                                          -------- Symbols --------      Pages       Processing
;       File                               Total    Loaded   Percent      Mapped      Time

;  _$255$DUA28:[SYSLIB]LIB.L32;1           18619       103         0      1000          00:01.8


;                                       COMMAND QUALIFIERS

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

; Size:          598 code + 8 data bytes
; Run Time:          00:16.6
; Elapsed Time:      00:35.9
; Lines/CPU Min:      1874
; Lexemes/CPU-Min:  28439
; Memory Used:  142 pages
; Compilation Complete
```

LIBACP
LIS

CALLIMAGE
LIS

DISMOUMSG
LIS

LIBCLICAL
LIS

LIBCOMAT
LIS

FILEDFLT
LIS

LIBASSIGN
LIS

CORMSG
LIS

CVTATB
LIS

LIBCQUAL
LIS

INITMSG
LIS

CREMAPSEC
LIS