```
IIIIIIIII      NNN          NNN     SSSSSSSSSSS  TTTTTTTTTTTTTTT     AAAAAAAAA     LLL
IIIIIIIII      NNN          NNN     SSSSSSSSSSS  TTTTTTTTTTTTTTT     AAAAAAAAA     LLL
IIIIIIIII      NNN          NNN     SSSSSSSSSSS  TTTTTTTTTTTTTTT     AAAAAAAAA     LLL
   III         NNN          NNN  SSS                   TTT          AAA         AAA  LLL
   III         NNN          NNN  SSS                   TTT          AAA         AAA  LLL
   III         NNN          NNN  SSS                   TTT          AAA         AAA  LLL
   III         NNNNNN       NNN  SSS                   TTT          AAA         AAA  LLL
   III         NNNNNN       NNN  SSS                   TTT          AAA         AAA  LLL
   III         NNNNNN       NNN  SSS                   TTT          AAA         AAA  LLL
   III         NNN    NNN   NNN     SSSSSSSSS          TTT          AAA         AAA  LLL
   III         NNN    NNN   NNN     SSSSSSSSS          TTT          AAA         AAA  LLL
   III         NNN    NNN   NNN     SSSSSSSSS          TTT          AAA         AAA  LLL
   III         NNN       NNNNNN               SSS      TTT          AAAAAAAAAAAAAAA  LLL
   III         NNN       NNNNNN               SSS      TTT          AAAAAAAAAAAAAAA  LLL
   III         NNN       NNNNNN               SSS      TTT          AAAAAAAAAAAAAAA  LLL
   III         NNN          NNN               SSS      TTT          AAA         AAA  LLL
   III         NNN          NNN               SSS      TTT          AAA         AAA  LLL
   III         NNN          NNN               SSS      TTT          AAA         AAA  LLL
IIIIIIIII      NNN          NNN  SSSSSSSSSSS           TTT          AAA         AAA  LLLLLLLLLLLLLLL
IIIIIIIII      NNN          NNN  SSSSSSSSSSS           TTT          AAA         AAA  LLLLLLLLLLLLLLL
IIIIIIIII      NNN          NNN  SSSSSSSSSSS           TTT          AAA         AAA  LLLLLLLLLLLLLLL
```

```
IIIIII    NN       NN    SSSSSSSS  MM       MM    AAAAAA     IIIIII    NN       NN
IIIIII    NN       NN    SSSSSSSS  MM       MM    AAAAAA     IIIIII    NN       NN
  II      NN       NN  SS          MMMM   MMMM   AA    AA     II        NN       NN
  II      NN       NN  SS          MMMM   MMMM   AA    AA     II        NN       NN
  II      NNNN     NN  SS          MM  MM  MM   AA    AA     II        NNNN     NN
  II      NNNN     NN  SS          MM  MM  MM   AA    AA     II        NNNN     NN
  II      NN  NN   NN    SSSSSS    MM      MM   AA    AA     II        NN  NN   NN
  II      NN  NN   NN    SSSSSS    MM      MM   AA    AA     II        NN  NN   NN
  II      NN    NNNN          SS  MM      MM   AAAAAAAAAA     II        NN    NNNN
  II      NN    NNNN          SS  MM      MM   AAAAAAAAAA     II        NN    NNNN
  II      NN      NN          SS  MM      MM   AA    AA     II        NN      NN      ....
  II      NN      NN          SS  MM      MM   AA    AA     II        NN      NN      ....
IIIIII    NN      NN  SSSSSSSS    MM      MM   AA    AA   IIIIII    NN      NN      ....
IIIIII    NN      NN  SSSSSSSS    MM      MM   AA    AA   IIIIII    NN      NN      ....

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 INSMAIN (                      ! INSTALL Main Entry and Command processing
;     2          0002  0                 IDENT = 'V04-000',
;     3          0003  0                 ADDRESSING_MODE(EXTERNAL = GENERAL),
;     4          0004  0                 MAIN = INSTALL_START
;     5          0005  0                 ) =
;     6          0006  1 BEGIN
;     7          0007  1
;     8          0008  1 !
;     9          0009  1 !*****************************************************************************
;    10          0010  1 !*                                                                           *
;    11          0011  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                                  *
;    12          0012  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                    *
;    13          0013  1 !*  ALL RIGHTS RESERVED.                                                      *
;    14          0014  1 !*                                                                           *
;    15          0015  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED     *
;    16          0016  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE    *
;    17          0017  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER     *
;    18          0018  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY     *
;    19          0019  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY     *
;    20          0020  1 !*  TRANSFERRED.                                                              *
;    21          0021  1 !*                                                                           *
;    22          0022  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE     *
;    23          0023  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT     *
;    24          0024  1 !*  CORPORATION.                                                              *
;    25          0025  1 !*                                                                           *
;    26          0026  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS     *
;    27          0027  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                   *
;    28          0028  1 !*                                                                           *
;    29          0029  1 !*                                                                           *
;    30          0030  1 !*****************************************************************************
;    31          0031  1
;    32          0032  1 !++
;    33          0033  1 ! FACILITY:  Install
;    34          0034  1 !
;    35          0035  1 ! ABSTRACT:
;    36          0036  1 !
;    37          0037  1 !       This module controls command parsing and processing.
;    38          0038  1 !
;    39          0039  1 ! ENVIRONMENT:
;    40          0040  1 !
;    41          0041  1 !       VAX/VMS operating system.
;    42          0042  1 !
;    43          0043  1 ! AUTHOR:  Bob Grosso, June 1981
;    44          0044  1 !
;    45          0045  1 ! Modified by:
;    46          0046  1 !
;    47          0047  1 !       V03-015 MSH0061         Michael S. Harvey       5-Jul-1984
;    48          0048  1 !               Add support for EXEONLY images. Also, don't carry old
;    49          0049  1 !               PURGE context from a previous INSTALL action into a new
;    50          0050  1 !               CREATE command.
;    51          0051  1 !
;    52          0052  1 !       V03-014 MSH0060         Michael S. Harvey       3-Jul-1984
;    53          0053  1 !               Partially back off MSH0055 so that reasonable status
;    54          0054  1 !               values don't cause the utility to exit.
;    55          0055  1 !
;    56          0056  1 !       V03-013 MSH0057         Michael S. Harvey       27-Jun-1984
;    57          0057  1 !               Propagate /WRITEABLE attribute across REPLACE command
```

```
;  58          0058  1 !               when appropriate.
;  59          0059  1 !
;  60          0060  1 !       V03-012 MSH0055         Michael S. Harvey       30-May-1984
;  61          0061  1 !               Don't infinitely loop when an error occurs trying to
;  62          0062  1 !               get a command line.
;  63          0063  1 !
;  64          0064  1 !       V03-011 MSH0052         Michael S. Harvey       22-May-1984
;  65          0065  1 !               Prevent deleting wrong KFE by ensuring that the KFE
;  66          0066  1 !               pointer contents is correct for all cases when opening
;  67          0067  1 !               the image file.
;  68          0068  1 !
;  69          0069  1 !       V03-010 MSH0048         Michael S. Harvey       15-May-1984
;  70          0070  1 !               Prevent some SPRs by adding some common sense to the
;  71          0071  1 !               status reporting for the REPLACE command.
;  72          0072  1 !
;  73          0073  1 !       V03-009 MSH0031         Michael S. Harvey       12-Apr-1984
;  74          0074  1 !               Ensure that a known file installed /OPEN is completely
;  75          0075  1 !               mapped by the retrieval pointers.
;  76          0076  1 !
;  77          0077  1 !       V03-008 MSH0025         Michael S. Harvey       2-Apr-1984
;  78          0078  1 !               Homogenize error reporting to prevent confusion (SPRs).
;  79          0079  1 !
;  80          0080  1 !       V03-007 MSH0022         Michael S. Harvey       15-Mar-1984
;  81          0081  1 !               Fully resolve device specifications by unconcealing
;  82          0082  1 !               the device. Eliminate middle directory brackets for
;  83          0083  1 !               root directories.
;  84          0084  1 !
;  85          0085  1 !       V03-006 MSH0015         Michael S. Harvey       5-Mar-1984
;  86          0086  1 !               Be a bit more intelligent when displaying other errors.
;  87          0087  1 !
;  88          0088  1 !       V03-005 MSH0009         Michael S. Harvey       16-Feb-1984
;  89          0089  1 !               Don't report errors that DCLPARSE already reported.
;  90          0090  1 !
;  91          0091  1 !       V03-004 BLS0256         Benn Schreiber          27-Dec-1983
;  92          0092  1 !               Allocate listing buffer once here. Use util$report_io_error
;  93          0093  1 !               and util$getfilename.  Convert KFD directory specs
;  94          0094  1 !               to all have angle brackets (square brackets not multinational).
;  95          0095  1 !               Make DELETE work even if file has been deleted.
;  96          0096  1 !
;  97          0097  1 !       V03-003 BLS0245         Benn Schreiber          27-Oct-1983
;  98          0098  1 !               Correct error severity for INS$_FAIL message to be
;  99          0099  1 !               WARNING for /LIST and /DELETE.
; 100          0100  1 !
; 101          0101  1 !       V03-002 RPG0002         Bob Grosso              26-Aug-1983
; 102          0102  1 !               Add a condition handler to record most severe error.
; 103          0103  1 !               Comment code more thoroughly.
; 104          0104  1 !
; 105          0105  1 !       V03-001 RPG0001         Bob Grosso              24-Aug-1983
; 106          0106  1 !               Check for Writeable qualifier before opening file
; 107          0107  1 !               so that /WRITE works.
; 108          0108  1 !               Re-activate header residency so that /HEAD works.
; 109          0109  1 !
; 110          0110  1 !-
```

```
;  112          0111  1 %SBTTL  'Declarations';
;  113          0112  1
;  114          0113  1 !
;  115          0114  1 ! Include files
;  116          0115  1 !
;  117          0116  1
;  118          0117  1 LIBRARY 'SYS$LIBRARY:LIB';              ! VAX/VMS system definitions
;  119          0118  1
;  120          0119  1 EXTERNAL ROUTINE
;  121          0120  1     CLI$DCL_PARSE,                      ! Parse DCL command
;  122          0121  1     CLI$DISPATCH,                       ! Dispatch to verb routines
;  123          0122  1     CLI$PRESENT,                        ! Check if entity present
;  124          0123  1     CLI$GET_VALUE,                      ! Get value from command line
;  125          0124  1     LBR$OUTPUT_HELP,                    ! print interactive help
;  126          0125  1     LIB$GET_FOREIGN,                    ! get command line
;  127          0126  1     LIB$GET_INPUT,                      ! Get subcommands
;  128          0127  1     LIB$GET_VM,                         ! Allocate virtual memory
;  129          0128  1     LIB$PUT_OUTPUT,                     ! print help text
;  130          0129  1     PRV$SETPRIV,                        ! set privilege bits given ASCII string
;  131          0130  1     STR$COMPARE,                        ! compare two strings
;  132          0131  1     STR$CONCAT,                         ! concatenate strings
;  133          0132  1     STR$COPY_DX,                        ! Copy strings by descriptor
;  134          0133  1     STR$POSITION,                       ! Find a substring on the command line
;  135          0134  1     STR$RIGHT,                          ! truncate part of command line
;  136          0135  1     STR$UPCASE,                         ! Raise case of command line
;  137          0136  1     UTIL$GETFILENAME,                   ! Get descriptor of file
;  138          0137  1     SYS$GETDVIW : ADDRESSING_MODE (GENERAL); ! Get Device Information
;  139          0138  1
;  140          0139  1 EXTERNAL LITERAL
;  141          0140  1     CLI$_NEGATED,                       ! Qualifier was explicitly negated
;  142          0141  1     CLI$_NOCOMD,                        ! Empty command line
;  143          0142  1     CLI$_NOQUAL,                        ! No qualifier permitted on command line
;  144          0143  1     PRV$_INVNAM,                        ! Invalid privilege name
;  145          0144  1     PRV$_NOTUNQ;                        ! Privilege name not unique
;  146          0145  1
;  147          0146  1 EXTERNAL LITERAL
;  148          0147  1     INS$C_FAOBUFLEN,                    ! Size of output buffer
;  149          0148  1     EXE$C_SYSEFN;                       ! System event flag, used when locking
;  150          0149  1
;  151          0150  1 EXTERNAL
;  152          0151  1     INS$FAOOUTBUF,                      ! Listing output buffer
;  153          0152  1     INS$FAOBUFDESC : $BBLOCK,           ! Descriptor of output buffer
;  154          0153  1     CTL$GQ_PROCPRIV : $BBLOCK,          ! privilege mask of user
;  155          0154  1     EXE$GQ_KFE_LCKNAM,                  ! Name of lock for Known file entries
;  156          0155  1     EXE$GL_SYSID_LOCK;                  ! System lock for KFE parent lock to
;  157          0156  1                                         !  avoid a trip down the CI on cluster systems
;  158          0157  1
;  159          0158  1 BIND
;  160          0159  1     INS$Q_LCKNAM =      $DESCRIPTOR ('INSTALL$INSLOCK');
;  161          0160  1
```

```
;  163          0161  1
;  164          0162  1 REQUIRE 'SRC$:INSPREFIX.REQ';
;  165          0304  1 REQUIRE 'LIB$:INSDEF.R32';
;  166          0363  1
;  167          0364  1 EXTERNAL
;  168          0365  1     INSCMD;                             ! Tables describing the Install commands
;  169          0366  1
;  170          0367  1 EXTERNAL LITERAL
;  171          0368  1     INS$_DELETED,                       ! Previous Known File Entry deleted
;  172          0369  1     INS$_EXISTS,                        ! Known File Entry already exists
;  173          0370  1     INS$_FAIL,                          ! Unable to <OPERATION> <FILENAME>
;  174          0371  1     INS$_FAILED,                        ! Failed to process request for <FILENAME>
;  175          0372  1     INS$_FAILGETVM,                     ! Failed to get virtual memory
;  176          0373  1     INS$_HELP,                          ! Enter /HELP for INSTALL help
;  177          0374  1     INS$_INTRNLERR,                     ! Internal Error: error detail
;  178          0375  1     INS$_INVPRVNAM,                     ! Invalid privilege name
;  179          0376  1     INS$_NOKFEFND,                      ! Known File not found
;  180          0377  1     INS$_NOLOAD,                        ! Network and sequentially loaded files are not installable
;  181          0378  1     INS$_NOPREV,                        ! No KFE deleted during REPLACE operation
;  182          0379  1     INS$_PRVNOTUNQ,                     ! Privilege name not unique
;  183          0380  1     INS$_REMOVED;                       ! Entry for deleted file has been removed
;  184          0381  1
;  185          0382  1 !
;  186          0383  1 ! External routines
;  187          0384  1 !
;  188          0385  1
;  189          0386  1 EXTERNAL ROUTINE
;  190          0387  1
;  191          0388  1 !   Routine             Defined in      Function
;  192          0389  1
;  193          0390  1     INS$PARSE_OLD_CMD,  ! INSOLDCMD     Support the crusty old interface
;  194          0391  1     INS$CREATE,         ! INSCREATE     perform CREATE command
;  195          0392  1     INS$DELETE,         ! INSDELETE     perform DELETE command
;  196          0393  1     INS$PURGE,          ! INSDELET      perform the PURGE command
;  197          0394  1     INS$GLOBAL,         ! INSGLOBAL     list global sections
;  198          0395  1     INS$LIST;           ! INSLIST       list known file entries
;  199          0396  1
;  200          0397  1 !
;  201          0398  1 ! Table of contents
;  202          0399  1 !
;  203          0400  1
;  204          0401  1 FORWARD ROUTINE
;  205          0402  1     PARSE_NEW_CMD,                      ! Call the new Dispatch routines
;  206          0403  1     INS$HELP_VERB,                      ! Perform HELP function
;  207          0404  1     INS$EXECUTE_IN_EXEC_WITH_R_LOCK,    ! Execute in exec mode to take out read lock and then execute routin
;  208          0405  1     INS$EXECUTE_IN_KRNL_WITH_W_LOCK,    ! Execute in kernel mode to take out write lock and then execute rou
;  209          0406  1     EXECUTE_WITH_R_LOCK,                ! Take out read lock and then execute routine
;  210          0407  1     EXECUTE_WITH_W_LOCK,                ! Take out write lock and then execute routine
;  211          0408  1     INS$CNVRT_KF_LOCK,                  ! Convert the mode of the KF lock
;  212          0409  1     INS$CVT_DIR : NOVALUE,              ! Convert and compress directory brackets
;  213          0410  1     INS$OPEN_FILE,                      ! handle file open processing
;  214          0411  1     INS$CLOSE_FILE,                     ! handle file close
;  215          0412  1     INS$CHECK_PRIV,                     ! enforce privilege check in case INSTALL is installed
;  216          0413  1     INS_HANDLER;                        ! Handler to pick up highest severity
;  217          0414  1
;  218          0415  1 GLOBAL
;  219          0416  1     SGN$GB_KFHSHSIZ,                    ! aa Remove
```

```
;  220          0417  1      INS$L_INTRNLERR,                    ! Pass back address of internal error descriptor
;  221          0418  1      INS$G_KFENAM : $NAM_DECL,           ! file name block declaration
;  222          0419  1      INS$GL_KFECHAN,                     ! channel on which known file is open
;  223          0420  1      INS$GL_CTLMSK : BLOCK [1],          ! Control flags resulting from command parse
;  224          0421  1      INS$GL_REPLACE_MSK: BLOCK [1],      ! Control flags saved for REPLACE
;  225          0422  1      INS$GQ_KFERNS :                     ! buffer for resultant name
;  226          0423  1          $BBLOCK [DSC$C_S_BLN],
;  227          0424  1      INS$GL_KFEADR,                      ! Address of known file entry
;  228          0425  1      INS$GQ_KFEPRIVS : $BBLOCK [8],      ! Quadword privilege mask resulting from command parse
;  229          0426  1      INS$GQ_FILDSC : DYN_DESC_DECL,      ! descriptor to obtain filename
;  230          0427  1                                          !  returned during command parse
;  231          0428  1
;  232          0429  1      INS$G_OUTRAB : $RAB_DECL,           ! Used for output by INS$LIST and
;  233          0430  1      INS$G_OUTFAB : $FAB_DECL,           !   INS$GLOBAL
;  234          0431  1      INS$G_KFEFAB : $FAB_DECL,           ! file access block declaration
;  235          0432  1      INS$G_KF_LKSB : $BBLOCK [8],        ! Known file entries Lock status block
;  236          0433  1      INS$G_INS_LKSB : $BBLOCK [8];       ! INSTALL lock status block
;  237          0434  1
;  238          0435  1  OWN
;  239          0436  1      INS$EXIT_STATUS : $BBLOCK [4]       ! Store worst error status to exit with
;  240          0437  1          INITIAL (SS$_NORMAL),
;  241          0438  1      PARSE_CMD_DESC : DYN_DESC_DECL,     ! descriptor of "INSTALL>" concatenated with input string
;  242          0439  1      INPUT_DESC : DYN_DESC_DECL,         ! command string input
;  243          0440  1      VERB_DESC : DYN_DESC_DECL,          !
;  244          0441  1      HELP_DESC : DYN_DESC_DECL,          !
;  245          0442  1      PRIV_DSC : DYN_DESC_DECL,           ! For parsing privileges on command line
;  246          0443  1      INTERACTIVE,                        ! boolean to record if this is a single execution of a
;  247          0444  1                                          !  foreign command or an interactive session.
;  248          0445  1      COMMAND_MODE,                       ! Boolean to recall which CLD to use
;  249          0446  1
;  250          0447  1  LITERAL
;  251          0448  1      NEW = 1,                            ! use New command verb interface
;  252          0449  1      OLD = 0;                            ! use old command qualifier interface
;  253          0450  1
;  254          0451  1  !
;  255          0452  1  ! Define shared messages for local use
;  256          0453  1  !
;  257        P 0454  1  $SHR_MSGDEF(INS,123,LOCAL,
;  258          0455  1          (OPENIN,ERROR));
;  259          0456  1
;  260          0457  1  BIND
;  261          0458  1      ! Descriptors of command strings used during command parseing
;  262          0459  1
;  263          0460  1      COMMAND_MODE_DSC =  $DESCRIPTOR ('COMMAND_MODE'),
;  264          0461  1      SLASH_COMM_DSC =    $DESCRIPTOR ('/COMM'),
;  265          0462  1
;  266          0463  1      HEL_DSC =           $DESCRIPTOR ('HEL'),
;  267          0464  1      HELP_DSC =          $DESCRIPTOR ('HELP'),
;  268          0465  1
;  269          0466  1      FILE_SPEC_DSC =     $DESCRIPTOR ('FILE_SPEC'),
;  270          0467  1
;  271          0468  1      CREATE_DSC =        $DESCRIPTOR ('CREATE'),
;  272          0469  1      REPLACE_DSC =       $DESCRIPTOR ('REPLACE'),
;  273          0470  1      DELETE_DSC =        $DESCRIPTOR ('DELETE'),
;  274          0471  1      REMOVE_DSC =        $DESCRIPTOR ('REMOVE'),
;  275          0472  1      LIST_DSC =          $DESCRIPTOR ('LIST'),
;  276          0473  1      LOG_DSC =           $DESCRIPTOR ('LOG'),
```

```
;  277     0474  1     GLOBAL_DSC =        $DESCRIPTOR ('GLOBAL'),
;  278     0475  1
;  279     0476  1     PRIVILEGED_DSC =    $DESCRIPTOR ('PRIVILEGED'),
;  280     0477  1     HDRRES_DSC =        $DESCRIPTOR ('HEADER_RESIDENT'),
;  281     0478  1     PROCESS_DSC =       $DESCRIPTOR ('PROCESS'),
;  282     0479  1     SHARED_DSC =        $DESCRIPTOR ('SHARED'),
;  283     0480  1     OPEN_DSC =          $DESCRIPTOR ('OPEN'),
;  284     0481  1     PROTECT_DSC =       $DESCRIPTOR ('PROTECTED'),
;  285     0482  1     WRITE_DSC =         $DESCRIPTOR ('WRITEABLE'),
;  286     0483  1     PURGE_DSC =         $DESCRIPTOR ('PURGE'),
;  287     0484  1     ACCOUNT_DSC =       $DESCRIPTOR ('ACCOUNTING'),
;  288     0485  1     XONLY_DSC =         $DESCRIPTOR ('EXECUTE_ONLY');
;  289     0486  1
```

```
;  291     0487  1 %SBTTL  'INSTALL_START';
;  292     0488  1
;  293     0489  1 GLOBAL ROUTINE  INSTALL_START =
;  294     0490  2 BEGIN
;  295     0491  2 !+++
;  296     0492  2 !
;  297     0493  2 !   FUNCTIONAL DESCRIPTION:
;  298     0494  2 !
;  299     0495  2 !        Check if there is a foreign command.
;  300     0496  2 !        If there is, see if it is preceded by /COMMAND_MODE which places INSTALL
;  301     0497  2 !        into the new command verb mode.  Then if /COMMAND_MODE is followed by
;  302     0498  2 !        a command, executed it as a single foreign command and terminate.
;  303     0499  2 !        If /COMMAND_MODE specified alone, execute in command prompting mode
;  304     0500  2 !        and parse for new command verbs.
;  305     0501  2 !
;  306     0502  2 !        If no /COMMAND_MODE present on foreign command line, then parse using
;  307     0503  2 !        the trashy old command qualifier interface.  Execute a foreign and quit
;  308     0504  2 !        or drop into prompt mode if no foreign command.
;  309     0505  2 !
;  310     0506  2 !   IMPLICIT INPUT:
;  311     0507  2 !
;  312     0508  2 !        none
;  313     0509  2 !
;  314     0510  2 !   OUTPUT:
;  315     0511  2 !
;  316     0512  2 !        Print command line prompt and signal any errors.
;  317     0513  2 !
;  318     0514  2 !   IMPLICIT OUTPUT:
;  319     0515  2 !
;  320     0516  2 !        none
;  321     0517  2 !
;  322     0518  2 !   ROUTINE VALUE:
;  323     0519  2 !
;  324     0520  2 !        Low bit set for success.  All errors are signalled.
;  325     0521  2 !
;  326     0522  2 !---
;  327     0523  2 LOCAL
;  328     0524  2     DDTSTR : BBLOCK [NAM$C_MAXRSS],
;  329     0525  2     STATUS;
;  330     0526  2
;  331     0527  2 BUILTIN
;  332     0528  2     FP;
;  333     0529  2
;  334     0530  2 .FP = INS_HANDLER;                      !Set handler
;  335     0531  2
;  336     0532  2 !+++
;  337     0533  2 !
;  338     0534  2 !   Initialization
;  339     0535  2 !
;  340     0536  2 !---
;  341     0537  2
;  342     0538  2 INS$GQ_KFERNS [DSC$A_POINTER] = DDTSTR;         ! Initialize DDT string pointer
;  343     0539  2
;  344     0540  2 COMMAND_MODE = OLD;
;  345     0541  2
;  346     0542  2
;  347     0543  2 SGN$GB_KFHSHSIZ = 128;                          ! *** Replace with a literal ***
```

```
;  348          0544  2
;  349        P 0545  2 $FAB_INIT(                                    ! Initialize the File Access Block
;  350        P 0546  2           FAB = INS$G_OUTFAB,
;  351        P 0547  2           FAC = PUT,
;  352        P 0548  2           RAT = CR,
;  353        P 0549  2           FNM = 'SYS$OUTPUT'
;  354          0550  2           );
;  355        P 0551  2 $RAB_INIT(                                    ! Initialize the Record Access Block
;  356        P 0552  2           RAB = INS$G_OUTRAB,
;  357        P 0553  2           FAB = INS$G_OUTFAB
;  358          0554  2           );
;  359          0555  2 REPORT ($CREATE (FAB = INS$G_OUTFAB) );         ! Create a channel to sys$output
;  360          0556  2 REPORT ($CONNECT (RAB = INS$G_OUTRAB) );        ! Connect to the channel
;  361          0557  2
;  362          0558  2 !
;  363          0559  2 ! Allocate fao output buffer
;  364          0560  2 !
;  365          0561  3 IF NOT (STATUS = LIB$GET_VM(%REF(INS$C_FAOBUFLEN), INS$FAOOUTBUF))
;  366          0562  2     THEN RETURN SIGNAL(INS$_FAILGETVM,1,INS$C_FAOBUFLEN,.STATUS);
;  367          0563  2
;  368          0564  2 !+++
;  369          0565  2 !
;  370          0566  2 !   See if there is a foreign command line.  If there is
;  371          0567  2 !   and it is not blank then flag non-interactive mode
;  372          0568  2 !   and execute only that command.
;  373          0569  2 !
;  374          0570  2 !---
;  375          0571  2
;  376          0572  2 INTERACTIVE = TRUE;                            ! Will loop for command input
;  377          0573  2
;  378          0574  2 REPORT (LIB$GET_FOREIGN( INPUT_DESC ) );
;  379          0575  2 IF ( .INPUT_DESC [DSC$W_LENGTH] NEQ 0 ) AND    ! If there is something on the line
;  380          0576  2    NOT CH$FAIL( CH$FIND_NOT_CH                  ! and there is something besides blanks
;  381          0577  2         ( .INPUT_DESC[DSC$W_LENGTH],
;  382          0578  2         .INPUT_DESC[DSC$A_POINTER], %C' '))
;  383          0579  2 THEN
;  384          0580  3     BEGIN
;  385          0581  3     LOCAL
;  386          0582  3         CMD_PTR,
;  387          0583  3         CMD_LEN;
;  388          0584  3     MAP
;  389          0585  3         COMMAND_MODE_DSC : $BBLOCK;
;  390          0586  3
;  391          0587  3     STR$UPCASE ( PARSE_CMD_DESC, INPUT_DESC );  ! Upcase and copy to PARSE_CMD_DESC
;  392          0588  3
;  393          0589  3     !+++
;  394          0590  3     !
;  395          0591  3     !   Check for /COMMAND_MODE
;  396          0592  3     !   If present by itself, then switch to new Command Language Definition
;  397          0593  3     !   and go interactive.
;  398          0594  3     !   If present, then switch to new Command Language Definition then execute
;  399          0595  3     !   the remainder of the command line as a single command and exit.
;  400          0596  3     !
;  401          0597  3     !
;  402          0598  3     !---
;  403          0599  3
;  404          0600  3     CMD_PTR = STR$POSITION ( PARSE_CMD_DESC, SLASH_COMM_DSC);   ! Look for "/COMM" on COMMAND_MODE line
```

```
;  405     0601  3      IF .CMD_PTR NEQ 0                                                ! If found
;  406     0602  3      THEN
;  407     0603  4          BEGIN
;  408     0604  4          LOCAL
;  409     0605  4              PC_PTR : REF $BBLOCK,
;  410     0606  4              SC_PTR : REF $BBLOCK;
;  411     0607  4
;  412     0608  4          !
;  413     0609  4          !   Remove as much of "/COMMAND_MODE" as was present
;  414     0610  4          !
;  415     0611  4          PC_PTR = .PARSE_CMD_DESC [DSC$A_POINTER] + 5;
;  416     0612  4          SC_PTR = .COMMAND_MODE_DSC [DSC$A_POINTER] + 4;
;  417     0613  4          CMD_LEN = 5;
;  418     0614  4
;  419     0615  4          WHILE .(.PC_PTR) <0, 8> EQL .(.SC_PTR) <0, 8> AND
;  420     0616  4                .CMD_LEN LSS .PARSE_CMD_DESC [DSC$W_LENGTH]
;  421     0617  4          DO
;  422     0618  5              BEGIN
;  423     0619  5              PC_PTR = .PC_PTR + 1;
;  424     0620  5              SC_PTR = .SC_PTR + 1;
;  425     0621  5              CMD_LEN = .CMD_LEN + 1;
;  426     0622  4              END;
;  427     0623  4
;  428     0624  4          !
;  429     0625  4          !   We've marched to the end of however much of "/COMMAND_MODE" had
;  430     0626  4          !   been entered.  Now drop that part and if anything is left
;  431     0627  4          !   treat it as a single command and then exit, otherwise, mark
;  432     0628  4          !   new CLD and proceed as though interactive, i.e. no foreign command.
;  433     0629  4          !
;  434     0630  4          STR$RIGHT ( PARSE_CMD_DESC, PARSE_CMD_DESC, %REF (.CMD_PTR+.CMD_LEN) );
;  435     0631  4          COMMAND_MODE = NEQ;
;  436     0632  4
;  437     0633  4          IF ( .PARSE_CMD_DESC [DSC$W_LENGTH] NEQ 0 ) AND ! If there is something on the line
;  438     0634  4             NOT CH$FAIL( CH$FIND_NOT_CH                   ! and there is something besides blanks
;  439     0635  4                  ( .PARSE_CMD_DESC [DSC$W_LENGTH],
;  440     0636  4                  .PARSE_CMD_DESC [DSC$A_POINTER], %C' '))
;  441     0637  4          THEN
;  442     0638  4              INTERACTIVE = FALSE;                          ! There was something on line other than /COMMAND_MO
;  443     0639  4                                                            !  so execute the rest and then quit
;  444     0640  4          END                                               ! /COMMAND_MODE was entered as foreign command
;  445     0641  3      ELSE
;  446     0642  3          INTERACTIVE = FALSE;                              ! Foreign command was present but did not include /C
;  447     0643  2      END;
;  448     0644  2
;  449     0645  2  INS$L_INTRNLERR = 0;                                     ! clear internal error descriptor
;  450     0646  2
;  451     0647  2  IF .COMMAND_MODE EQL OLD
;  452     0648  2  THEN
;  453     0649  2      INS$PARSE_OLD_CMD (PARSE_CMD_DESC, .INTERACTIVE)
;  454     0650  2  ELSE
;  455     0651  2      PARSE_NEW_CMD ();
;  456     0652  2
;  457     0653  2  RETURN (STATUS = .INS$EXIT_STATUS OR STS$M_INHIB_MSG);   ! Return worst status
;  458     0654  2
;  459     0655  1  END;                ! Global routine install_start
```

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

                                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

4B  43  4F  4C  53  4E  49  24  4C  4C  41  54  53  4E  49  00000 P.AAB:   .ASCII  \INSTALL$INSLOCK\                ;
                                                            0000F       .BLKB   1
                                                        0000000F  00010 P.AAA:   .LONG   15                       ;
                                                        00000000' 00014         .ADDRESS P.AAB                  ;
            45  44  4F  4D  5F  44  4E  41  4D  4D  4F  43  00018 P.AAD:   .ASCII  \COMMAND_MODE\               ;
                                                        0000000C  00024 P.AAC:   .LONG   12                       ;
                                                        00000000' 00028         .ADDRESS P.AAD                  ;
                                                    4D  4D  4F  43  2F  0002C P.AAF:   .ASCII  \/COMM\               ;
                                                                00031       .BLKB   3
                                                        00000005  00034 P.AAE:   .LONG   5                        ;
                                                        00000000' 00038         .ADDRESS P.AAF                  ;
                                                            4C  45  48  0003C P.AAH:   .ASCII  \HEL\                ;
                                                                0003F       .BLKB   1
                                                        00000003  00040 P.AAG:   .LONG   3                        ;
                                                        00000000' 00044         .ADDRESS P.AAH                  ;
                                                        50  4C  45  48  00048 P.AAJ:   .ASCII  \HELP\               ;
                                                        00000004  0004C P.AAI:   .LONG   4                        ;
                                                        00000000' 00050         .ADDRESS P.AAJ                  ;
                        43  45  50  53  5F  45  4C  49  46  00054 P.AAL:   .ASCII  \FILE_SPEC\              ;
                                                                0005D       .BLKB   3
                                                        00000009  00060 P.AAK:   .LONG   9                        ;
                                                        00000000' 00064         .ADDRESS P.AAL                  ;
                                    45  54  41  45  52  43  00068 P.AAN:   .ASCII  \CREATE\                ;
                                                                0006E       .BLKB   2
                                                        00000006  00070 P.AAM:   .LONG   6                        ;
                                                        00000000' 00074         .ADDRESS P.AAN                  ;
                                45  43  41  4C  50  45  52  00078 P.AAP:   .ASCII  \REPLACE\                ;
                                                                0007F       .BLKB   1
                                                        00000007  00080 P.AAO:   .LONG   7                        ;
                                                        00000000' 00084         .ADDRESS P.AAP                  ;
                                    45  54  45  4C  45  44  00088 P.AAR:   .ASCII  \DELETE\                ;
                                                                0008E       .BLKB   2
                                                        00000006  00090 P.AAQ:   .LONG   6                        ;
                                                        00000000' 00094         .ADDRESS P.AAR                  ;
                                    45  56  4F  4D  45  52  00098 P.AAT:   .ASCII  \REMOVE\                ;
                                                                0009E       .BLKB   2
                                                        00000006  000A0 P.AAS:   .LONG   6                        ;
                                                        00000000' 000A4         .ADDRESS P.AAT                  ;
                                                        54  53  49  4C  000A8 P.AAV:   .ASCII  \LIST\               ;
                                                        00000004  000AC P.AAU:   .LONG   4                        ;
                                                        00000000' 000B0         .ADDRESS P.AAV                  ;
                                                            47  4F  4C  000B4 P.AAX:   .ASCII  \LOG\                ;
                                                                000B7       .BLKB   1
                                                        00000003  000B8 P.AAW:   .LONG   3                        ;
                                                        00000000' 000BC         .ADDRESS P.AAX                  ;
                                    4C  41  42  4F  4C  47  000C0 P.AAZ:   .ASCII  \GLOBAL\                ;
                                                                000C6       .BLKB   2
                                                        00000006  000C8 P.AAY:   .LONG   6                        ;
                                                        00000000' 000CC         .ADDRESS P.AAZ                  ;
44  45  47  45  4C  49  56  49  52  50  000D0 P.ABB:   .ASCII  \PRIVILEGED\            ;
                                                                000DA       .BLKB   2
                                                        0000000A  000DC P.ABA:   .LONG   10                       ;
```

```
                                00000000'  000E0          .ADDRESS P.ABB
54 4E 45 44 49 53 45 52 5F 52 45 44 41 45 48  000E4 P.ABD:  .ASCII  \HEADER_RESIDENT\
                                           000F3          .BLKB   1
                                0000000F   000F4 P.ABC:   .LONG   15
                                00000000'  000F8          .ADDRESS P.ABD
                          53 53 45 43 4F 52 50  000FC P.ABF:  .ASCII  \PROCESS\
                                           00103          .BLKB   1
                                00000007   00104 P.ABE:   .LONG   7
                                00000000'  00108          .ADDRESS P.ABF
                             44 45 52 41 48 53  0010C P.ABH:  .ASCII  \SHARED\
                                           00112          .BLKB   2
                                00000006   00114 P.ABG:   .LONG   6
                                00000000'  00118          .ADDRESS P.ABH
                                   4E 45 50 4F  0011C P.ABJ:  .ASCII  \OPEN\
                                00000004   00120 P.ABI:   .LONG   4
                                00000000'  00124          .ADDRESS P.ABJ
                       44 45 54 43 45 54 4F 52 50  00128 P.ABL:  .ASCII  \PROTECTED\
                                           00131          .BLKB   3
                                00000009   00134 P.ABK:   .LONG   9
                                00000000'  00138          .ADDRESS P.ABL
                       45 4C 42 41 45 54 49 52 57  0013C P.ABN:  .ASCII  \WRITEABLE\
                                           00145          .BLKB   3
                                00000009   00148 P.ABM:   .LONG   9
                                00000000'  0014C          .ADDRESS P.ABN
                             45 47 52 55 50  00150 P.ABP:  .ASCII  \PURGE\
                                           00155          .BLKB   3
                                00000005   00158 P.ABO:   .LONG   5
                                00000000'  0015C          .ADDRESS P.ABP
                    47 4E 49 54 4E 55 4F 43 43 41  00160 P.ABR:  .ASCII  \ACCOUNTING\
                                           0016A          .BLKB   2
                                0000000A   0016C P.ABQ:   .LONG   10
                                00000000'  00170          .ADDRESS P.ABR
59 4C 4E 4F 5F 45 54 55 43 45 58 45  00174 P.ABT:  .ASCII  \EXECUTE_ONLY\
                                0000000C   00180 P.ABS:   .LONG   12
                                00000000'  00184          .ADDRESS P.ABT
         54 55 50 54 55 4F 24 53 59 53  00188 P.ABU:  .ASCII  \SYS$OUTPUT\

                                                        .PSECT  $OWN$,NOEXE,2

                                00000001   00000 INS$EXIT_STATUS:
                                                        .LONG   1
                                     00#   00004 PARSE_CMD_DESC:
                                                        .BYTE   0[2]
                                  02 0E    00006          .BYTE   14, 2
                                           00008          .BLKB   4
                                     00#   0000C INPUT_DESC:
                                                        .BYTE   0[2]
                                  02 0E    0000E          .BYTE   14, 2
                                           00010          .BLKB   4
                                     00#   00014 VERB_DESC:
                                                        .BYTE   0[2]
                                  02 0E    00016          .BYTE   14, 2
                                           00018          .BLKB   4
                                     00#   0001C HELP_DESC:
                                                        .BYTE   0[2]
                                  02 0E    0001E          .BYTE   14, 2
                                           00020          .BLKB   4
```

```
                00# 00024 PRIV_DSC:
                                .BYTE   0[2]
            02  0E  00026          .BYTE   14, 2
                00028           .BLKB   4
                0002C INTERACTIVE:
                                .BLKB   4
                00030 COMMAND_MODE:
                                .BLKB   4

                                .PSECT  $GLOBAL$,NOEXE,2

                00000 SGN$GB_KFHSHSIZ::
                                .BLKB   4
                00004 INS$L_INTRNLERR::
                                .BLKB   4
                00008 INS$G_KFENAM::
                                .BLKB   96
                00068 INS$GL_KFECHAN::
                                .BLKB   4
                0006C INS$GL_CTLMSK::
                                .BLKB   4
                00070 INS$GL_REPLACE_MSK::
                                .BLKB   4
                00074 INS$GQ_KFERNS::
                                .BLKB   8
                0007C INS$GL_KFEADR::
                                .BLKB   4
                00080 INS$GQ_KFEPRIVS::
                                .BLKB   8
                00# 00088 INS$GQ_FILDSC::
                                .BYTE   0[2]
            02  0E  0008A          .BYTE   14, 2
                0008C           .BLKB   4
                00090 INS$G_OUTRAB::
                                .BLKB   68
                000D4 INS$G_OUTFAB::
                                .BLKB   80
                00124 INS$G_KFEFAB::
                                .BLKB   80
                00174 INS$G_KF_LKSB::
                                .BLKB   8
                0017C INS$G_INS_LKSB::
                                .BLKB   8

                        INS$Q_LCKNAM=           P.AMA
                        COMMAND_MODE_DSC=       P.AAC
                        SLASH_COMM_DSC=         P.AAE
                        HEL_DSC=                P.AAG
                        HELP_DSC=               P.AAI
                        FILE_SPEC_DSC=          P.AAK
                        CREATE_DSC=             P.AAM
                        REPLACE_DSC=            P.AAO
                        DELETE_DSC=             P.AAQ
                        REMOVE_DSC=             P.AAS
                        LIST_DSC=               P.AAU
                        LOG_DSC=                P.AAW
                        GLOBAL_DSC=             P.AAY
```

```
                        PRIVILEGED_DSC=     P.ABA
                        HDRRES_DSC=         P.ABC
                        PROCESS_DSC=        P.ABE
                        SHARED_DSC=         P.ABG
                        OPEN_DSC=           P.ABI
                        PROTECT_DSC=        P.ABK
                        WRITE_DSC=          P.ABM
                        PURGE_DSC=          P.ABO
                        ACCOUNT_DSC=        P.ABQ
                        XONLY_DSC=          P.ABS
                        $RMS_PTR=           INS$G_OUTFAB
                        $RMS_PTR=           INS$G_OUTRAB
                                .EXTRN  CLI$DCL_PARSE, CLI$DISPATCH
                                .EXTRN  CLI$PRESENT, CLI$GET_VALUE
                                .EXTRN  LBR$OUTPUT_HELP
                                .EXTRN  LIB$GET_FOREIGN
                                .EXTRN  LIB$GET_INPUT, LIB$GET_VM
                                .EXTRN  LIB$PUT_OUTPUT, PRV$SETPRIV
                                .EXTRN  STR$COMPARE, STR$CONCAT
                                .EXTRN  STR$COPY_DX, STR$POSITION
                                .EXTRN  STR$RIGHT, STR$UPCASE
                                .EXTRN  UTIL$GETFILENAME
                                .EXTRN  SYS$GETDVIW, CLI$_NEGATED
                                .EXTRN  CLI$_NOCOMD, CLI$_NOQUAL
                                .EXTRN  PRV$_INVNAM, PRV$_NOTUNQ
                                .EXTRN  INS$C_FAOBUFLEN
                                .EXTRN  EXE$C_SYSEFN, INS$FAOOUTBUF
                                .EXTRN  INS$FAOBUFDESC, CTL$GQ_PROCPRIV
                                .EXTRN  EXE$GQ_KFE_LCKNAM
                                .EXTRN  EXE$GL_SYSID_LOCK
                                .EXTRN  INSCMD, INS$_DELETED
                                .EXTRN  INS$_EXISTS, INS$_FAIL
                                .EXTRN  INS$_FAILED, INS$_FAILGETVM
                                .EXTRN  INS$_HELP, INS$_INTRNLERR
                                .EXTRN  INS$_INVPRVNAM, INS$_NOKFEFND
                                .EXTRN  INS$_NOLOAD, INS$_NOPREV
                                .EXTRN  INS$_PRVNOTUNQ, INS$_REMOVED
                                .EXTRN  INS$PARSE_OLD_CMD
                                .EXTRN  INS$CREATE, INS$DELETE
                                .EXTRN  INS$PURGE, INS$GLOBAL
                                .EXTRN  INS$LIST, SYS$CREATE
                                .EXTRN  SYS$CONNECT

                                .PSECT  $CODE$,NOWRT,2

                                                03FC 00000  .ENTRY  INSTALL_START, Save R2,R3,R4,R5,R6,R7,R8,R9 ; 0489
                        59 00000000G    8F      D0 00002    MOVL    #INS$C_FAOBUFLEN, R9                        ;
                        58 00000000G    00      9E 00009    MOVAB   LIB$SIGNAL, R8                              ;
                        57      0000'   CF      9E 00010    MOVAB   $RMS_PTR, R7                                ;
                        56      0000'   CF      9E 00015    MOVAB   PARSE_CMD_DESC, R6                          ;
                        5E      FEFC    CE      9E 0001A    MOVAB   -260(SP), SP                                ;
                        6D      0000V   CF      9E 0001F    MOVAB   INS_HANDLER, (FP)                           ; 0530
                    A4  A7        04    AE      9E 00024    MOVAB   DDTSTR, INS$GQ_KFERNS+4                     ; 0538
                                  2C    A6      D4 00029    CLRL    COMMAND_MODE                                ; 0540
                    FF2C    C7    80    8F      9A 0002C    MOVZBL  #128, SGN$GB_KFHSHSIZ                       ; 0543
0050    8F      00          6E          0_      2C 00032    MOVC5   #0, (SP), #0, #80, $RMS_PTR                 ; 0550
                                                67     00039                                                    ;
```

```
                                67      5003    8F  B0 0003A           MOVW    #20483, $RMS_PTR                        ;
                             16 A7              01  90 0003F           MOVB    #1, $RMS_PTR+22                         ;
                             1E A7      0202    8F  B0 00043           MOVW    #514, $RMS_PTR+30                       ;
                             2C A7      0000'   CF  9E 00049           MOVAB   P.ABU, $RMS_PTR+44                      ;
                             34 A7              0A  90 0004F           MOVB    #10, $RMS_PTR+52                        ;
0044    8F          00                  6E          00  2C 00053       MOVC5   #0, (SP), #0, #68, $RMS_PTR             ; 0554
                                            BC  A7          0005A
                             BC A7      4401    8F  B0 0005C           MOVW    #17409, $RMS_PTR                        ;
                             F8 A7              67  9E 00062           MOVAB   INS$G_OUTFAB, $RMS_PTR+60               ;
                                                57  DD 00066           PUSHL   R7                                      ; 0555
                    00000000G 00                01  FB 00068           CALLS   #1, SYS$CREATE                          ;
                              05                50  E8 0006F           BLBS    STATUS, 1$                              ;
                                                50  DD 00072           PUSHL   STATUS                                  ;
                                68              01  FB 00074           CALLS   #1, LIB$SIGNAL                          ;
                                            BC  A7  9F 00077 1$:        PUSHAB  INS$G_OUTRAB                            ; 0556
                    00000000G 00                01  FB 0007A           CALLS   #1, SYS$CONNECT                         ;
                              05                50  E8 00081           BLBS    STATUS, 2$                              ;
                                                50  DD 00084           PUSHL   STATUS                                  ;
                                68              01  FB 00086           CALLS   #1, LIB$SIGNAL                          ;
                                        00000000G   00  9F 00089 2$:    PUSHAB  INS$FAOOUTBUF                           ; 0561
                             04 AE                  59  D0 0008F       MOVL    R9, 4(SP)                               ;
                                            04  AE  9F  00093           PUSHAB  4(SP)                                   ;
                    00000000G 00                02  FB 00096           CALLS   #2, LIB$GET_VM                          ;
                              54                50  D0 0009D           MOVL    R0, STATUS                              ;
                              10                54  E8 000A0           BLBS    STATUS, 3$                              ;
                                                54  DD 000A3           PUSHL   STATUS                                  ; 0562
                                                59  DD 000A5           PUSHL   R9                                      ;
                                                01  DD 000A7           PUSHL   #1                                      ;
                                        00000000G   8F  DD 000A9       PUSHL   #INS$_FAILGETVM                         ;
                                68              04  FB 000AF           CALLS   #4, LIB$SIGNAL                          ;
                                                    04 000B2           RET                                             ;
                             28 A6              01  D0 000B3 3$:       MOVL    #1, INTERACTIVE                         ; 0572
                                            08  A6  9F  000B7           PUSHAB  INPUT_DESC                              ; 0574
                    00000000G 00                01  FB 000BA           CALLS   #1, LIB$GET_FOREIGN                     ;
                              05                50  E8 000C1           BLBS    STATUS, 4$                              ;
                                                50  DD 000C4           PUSHL   STATUS                                  ;
                                68              01  FB 000C6           CALLS   #1, LIB$SIGNAL                          ;
                                50          08  A6  3C 000C9 4$:       MOVZWL  INPUT_DESC, R0                          ; 0575
                                                76  13 000CD           BEQL    10$                                     ;
0C      B6                      50              20  3B 000CF           SKPC    #32, R0, @INPUT_DESC+4                  ; 0577
                                                02  12 000D4           BNEQ    5$                                      ;
                                                51  D4 000D6           CLRL    R1                                      ;
                                                51  D5 000D8 5$:       TSTL    R1                                      ; 0578
                                                69  13 000DA           BEQL    10$                                     ;
                                            08  A6  9F  000DC           PUSHAB  INPUT_DESC                              ; 0587
                                                56  DD 000DF           PUSHL   R6                                      ;
                    00000000G 00                02  FB 000E1           CALLS   #2, STR$UPCASE                          ;
                                        0000'   CF  9F 000E8           PUSHAB  SLASH_COMM_DSC                          ; 0600
                                                56  DD 000EC           PUSHL   R6                                      ;
                    00000000G 00                02  FB 000EE           CALLS   #2, STR$POSITION                        ;
                                                50  D5 000F5           TSTL    CMD_PTR                                 ; 0601
                                                49  13 000F7           BEQL    9$                                      ;
                      53    04  A6                05  C1 000F9       ADDL3   #5, PARSE_CMD_DESC+4, PC_PTR            ; 0611
                      52  0000' CF                04  C1 000FE       ADDL3   #4, COMMAND_MODE_DSC+4, SC_PTR          ; 0612
                                                51  D0 00104           MOVL    #5, CMD_LEN                             ; 0613
                                                62  63 91 00107 6$:       CMPB    (PC_PTR), (SC_PTR)                      ; 0615
                                                0F  12 0010A           BNEQ    7$                                      ;
```

```
                51              66              10              00  ED 0010C           CMPZV   #0, #16, PARSE_CMD_DESC, CMD_LEN        ; 0616
                                                                08  15 00111           BLEQ    7$
                                                                53  D6 00113           INCL    PC_PTR                                  ; 0619
                                                                52  D6 00115           INCL    SC_PTR                                  ; 0620
                                                                51  D6 00117           INCL    CMD_LEN                                 ; 0621
                                                                EC  11 00119           BRB     6$                                      ; 0615
                                6E              50              51  C1 0011B 7$:       ADDL3   CMD_LEN, CMD_PTR, (SP)                  ; 0630
                                                        4040    8F  BB 0011F           PUSHR   #^M<R6,SP>
                                                                56  DD 00123           PUSHL   R6
                                    00000000G   00              03  FB 00125           CALLS   #3, STR$RIGHT
                                        2C      A6              01  D0 0012C           MOVL    #1, COMMAND_MODE                        ; 0631
                                                50              66  3C 00130           MOVZWL  PARSE_CMD_DESC, R0                      ; 0633
                                                                10  13 00133           BEQL    10$
                            04  B6              50              20  3B 00135           SKPC    #32, R0, @PARSE_CMD_DESC+4              ; 0635
                                                                02  12 0013A           BNEQ    8$
                                                                51  D4 0013C           CLRL    R1
                                                                51  D5 0013E 8$:       TSTL    R1                                      ; 0636
                                                                03  13 00140           BEQL    10$
                                                          28    A6  D4 00142 9$:       CLRL    INTERACTIVE                             ; 0642
                                                        FF30    C7  D4 00145 10$:      CLRL    INS$L_INTRNLERR                         ; 0645
                                                          2C    A6  D5 00149           TSTL    COMMAND_MODE                            ; 0647
                                                                0E  12 0014C           BNEQ    11$
                                                          28    A6  DD 0014E           PUSHL   INTERACTIVE                             ; 0649
                                                                56  DD 00151           PUSHL   R6
                                    00000000G   00              02  FB 00153           CALLS   #2, INS$PARSE_OLD_CMD
                                                                05  11 0015A           BRB     12$
                                        0000V   CF              00  FB 0015C 11$:      CALLS   #0, PARSE_NEW_CMD                       ; 0651
                                54      FC      A6      10000000    8F  C9 00161 12$:      BISL3   #268435456, INS$EXIT_STATUS, STATUS     ; 0653
                                                50              54  D0 0016A           MOVL    STATUS, R0
                                                                04  0016D           RET                                             ; 0655

; Routine Size:  366 bytes,    Routine Base:  $CODE$ + 0000

;  460          0656  1
```

```
;  462          0657  1 %SBTTL  'parse_new_cmd';
;  463          0658  1
;  464          0659  1 ROUTINE  PARSE_NEW_CMD =
;  465          0660  2 BEGIN
;  466          0661  2 !+++
;  467          0662  2 !
;  468          0663  2 !
;  469          0664  2 !---
;  470          0665  2 LOCAL
;  471          0666  2     STATUS;
;  472          0667  2
;  473          0668  2 IF NOT .INTERACTIVE
;  474          0669  2 THEN
;  475          0670  3     BEGIN
;  476          0671  3
;  477          0672  3     STATUS = CLI$DCL_PARSE( PARSE_CMD_DESC, INSCMD );   ! parse the command line
;  478          0673  3     IF .STATUS
;  479          0674  3     THEN
;  480          0675  3         REPORT (CLI$DISPATCH () );
;  481          0676  3     END
;  482          0677  3
;  483          0678  2 ELSE                                    ! INTERACTIVE
;  484          0679  3     BEGIN
;  485          0680  3     DO                                  ! Keep looping while still interactive
;  486          0681  4         BEGIN
;  487          0682  4
;  488          0683  4         DO STATUS = CLI$DCL_PARSE (0, INSCMD, LIB$GET_INPUT,
;  489          0684  4                         LIB$GET_INPUT, $DESCRIPTOR('INSTALL> ') )
;  490          0685  4         UNTIL .STATUS NEQ CLI$_NOCOMD;
;  491          0686  4
;  492          0687  4         IF .STATUS
;  493          0688  4         THEN
;  494          0689  5             BEGIN
;  495          0690  5             INS$L_INTRNLERR = 0;        ! clear internal error descriptor
;  496          0691  5             REPORT (CLI$DISPATCH ());
;  497          0692  5             END
;  498          0693  4         ELSE
;  499          0694  5             BEGIN
;  500          0695  5             IF .STATUS EQL RMS$_FNF THEN RETURN .STATUS;
;  501          0696  5             IF .STATUS EQL RMS$_EOF THEN RETURN TRUE;
;  502          0697  5             IF .STATUS EQL CLI$_NOQUAL
;  503          0698  5             THEN
;  504          0699  6                 BEGIN
;  505          0700  6                 !
;  506          0701  6                 !       If status is CLI$_NOQUAL then user may have typed
;  507          0702  6                 !       HELP /QUALIFIER and we really should drop into HELP
;  508          0703  6                 !       for him rather than barf out with a noqual message.
;  509          0704  6                 !
;  510          0705  6                 LOCAL
;  511          0706  6                     HELP_PTR;
;  512          0707  6
;  513          0708  6                 CLI$GET_VALUE (%ASCID '$LINE', VERB_DESC);
;  514          0709  6
;  515          0710  6                 HELP_PTR = STR$POSITION (VERB_DESC, HEL_DSC);
;  516          0711  6                 IF .HELP_PTR NEQ 0
;  517          0712  6                 THEN
;  518          0713  6                     INS$HELP_VERB ()
```

```
;  519          0714  6                 ELSE
;  520          0715  6                     SIGNAL (.STATUS);
;  521          0716  5                 END;
;  522          0717  4             END;
;  523          0718  4         END                             ! end of DO-WHILE interactive
;  524          0719  3     WHILE .INTERACTIVE;
;  525          0720  2     END;                                ! Interactive
;  526          0721  2
;  527          0722  2 RETURN TRUE;
;  528          0723  1 END;                                    ! Routine parse_new_cmd


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                    20  3E  4C  4C  41  54  53  4E  49  00192 P.ABW:  .ASCII  \INSTALL> \                          ;
                                                        0019B         .BLKB   1                                   ;
                                                    00000009  0019C P.ABV:  .LONG   9                               ;
                                                    00000000' 001A0         .ADDRESS P.ABW                      ;
                        00  00  00  45  4E  49  4C  24  001A4 P.ABY:  .ASCII  \$LINE\<0><0><0>                   ;
                                                    010E0005  001AC P.ABX:  .LONG   17694725                    ;
                                                    00000000' 001B0         .ADDRESS P.ABY                      ;


                                                                .PSECT  $CODE$,NOWRT,2

                                                        01FC 00000 PARSE_NEW_CMD:
                                                                   .WORD   Save R2,R3,R4,R5,R6,R7,R8            ; 0659
                                58 00000000G   00  9E 00002     MOVAB   LIB$GET_INPUT, R8                       ;
                                57 00000000G   00  9E 00009     MOVAB   LIB$SIGNAL, R7                          ;
                                56 00000000G   00  9E 00010     MOVAB   CLI$DISPATCH, R6                        ;
                                55 00000000G   00  9E 00017     MOVAB   CLI$DCL_PARSE, R5                       ;
                                54 00000000G   00  9E 0001E     MOVAB   INSCMD, R4                              ;
                                53      0000'  CF  9E 00025     MOVAB   INTERACTIVE, R3                         ;
                                1B             63  E8 0002A     BLBS    INTERACTIVE, 2$                         ; 0668
                                               54  DD 0002D     PUSHL   R4                                      ; 0672
                                        D8     A3  9F 0002F     PUSHAB  PARSE_CMD_DESC                          ;
                                65             02  FB 00032     CALLS   #2, CLI$DCL_PARSE                       ;
                                52             50  D0 00035     MOVL    R0, STATUS                              ;
                                0B             52  E9 00038     BLBC    STATUS, 1$                              ; 0673
                                66             00  FB 0003B     CALLS   #0, CLI$DISPATCH                        ; 0675
                                05             50  E8 0003E     BLBS    STATUS, 1$                              ;
                                               50  DD 00041     PUSHL   STATUS                                  ;
                                67             01  FB 00043     CALLS   #1, LIB$SIGNAL                          ;
                                               7A  11 00046 1$:  BRB     8$                                      ; 0668
                                        0000'  CF  9F 00048 2$:  PUSHAB  P.ABV                                   ; 0684
                                               58  DD 0004C     PUSHL   R8                                      ; 0683
                                        0110   8F  BB 0004E     PUSHR   #^M<R4,R8>                              ;
                                               7E  D4 00052     CLRL    -(SP)                                   ;
                                65             05  FB 00054     CALLS   #5, CLI$DCL_PARSE                       ;
                                52             50  D0 00057     MOVL    R0, STATUS                              ;
                    00000000G   8F             52  D1 0005A     CMPL    STATUS, #CLI$_NOCOMD                    ; 0685
                                               E5  13 00061     BEQL    2$                                      ;
                                0E             52  E9 00063     BLBC    STATUS, 3$                              ; 0687
                                        0000'  CF  D4 00066     CLRL    INS$L_INTRNLERR                         ; 0690
                                66             00  FB 0006A     CALLS   #0, CLI$DISPATCH                        ; 0691
```

```
                         4F            50  E8 0006D       BLBS    STATUS, 7$                                   ;
                                       50  DD 00070       PUSHL   STATUS                                       ;
                                       48  11 00072       BRB     6$                                           ;
                00018292  8F           52  D1 00074 3$:   CMPL    STATUS, #98962                               ; 0695
                                       04  12 0007B       BNEQ    4$                                           ;
                         50            52  D0 0007D       MOVL    STATUS, R0                                   ;
                                           04 00080       RET                                                  ;
                0001827A  8F           52  D1 00081 4$:   CMPL    STATUS, #98938                               ; 0696
                                       38  13 00088       BEQL    8$                                           ;
                00000000G 8F           52  D1 0008A       CMPL    STATUS, #CLI$_NOQUAL                         ; 0697
                                       2C  12 00091       BNEQ    7$                                           ;
                                 E8    A3  9F 00093       PUSHAB  VERB_DESC                                    ; 0708
                              0000'    CF  9F 00096       PUSHAB  P.ABX                                        ;
                00000000G 00           02  FB 0009A       CALLS   #2, CLI$GET_VALUE                            ;
                              0000'    CF  9F 000A1       PUSHAB  HEL_DSC                                      ; 0710
                                 E8    A3  9F 000A5       PUSHAB  VERB_DESC                                    ;
                00000000G 00           02  FB 000A8       CALLS   #2, STR$POSITION                             ;
                                       50  D5 000AF       TSTL    HELP_PTR                                     ; 0711
                                       07  13 000B1       BEQL    5$                                           ;
                    0000V CF           00  FB 000B3       CALLS   #0, INS$HELP_VERB                            ; 0713
                                       05  11 000B8       BRB     7$                                           ;
                                       52  DD 000BA 5$:   PUSHL   STATUS                                       ; 0715
                         67            01  FB 000BC 6$:   CALLS   #1, LIB$SIGNAL                               ;
                         86            63  E8 000BF 7$:   BLBS    INTERACTIVE, 2$                              ; 0719
                         50            01  D0 000C2 8$:   MOVL    #1, R0                                       ; 0722
                                           04 000C5       RET                                                  ; 0723

; Routine Size:  198 bytes,    Routine Base:  $CODE$ + 016E

;  529          0724  1
```

```
;  531          0725  1 %SBTTL  'INS$HELP_VERB';
;  532          0726  1 GLOBAL ROUTINE INS$HELP_VERB =
;  533          0727  1 !+++
;               0728  1 !
;  535          0729  1 !   New command interface action routine for HELP
;  536          0730  1 !   Request help be printed by lib$put_output to sys$output,
;  537          0731  1 !   from library SYS$HELP:INSTALHLP.HLB.  Query for additional help
;  538          0732  1 !   to sys$input using lib$get_input.
;  539          0733  1 !
;  540          0734  1 !---
;  541          0735  2 BEGIN
;  542          0736  2
;  543          0737  2 CLI$GET_VALUE (%ASCID 'HELP_LINE', HELP_DESC);
;  544          0738  2
;  545        P 0739  2 REPORT ( LBR$OUTPUT_HELP (LIB$PUT_OUTPUT, 0, HELP_DESC,
;  546          0740  2                         $DESCRIPTOR('INSTALHLP'), 0,LIB$GET_INPUT) );
;  547          0741  2 RETURN TRUE;
;  548          0742  1 END;                                       ! routine INS$HELP_VERB


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

          00 00 00 45 4E 49 4C 5F 50 4C 45 48  001B4 P.ACA:   .ASCII  \HELP_LINE\<0><0><0>                ;
                                               001C0 P.ABZ:   .LONG   17694729                             ;
                                               001C4          .ADDRESS P.ACA                               ;
                   50 4C 48 4C 41 54 53 4E 49  001C8 P.ACC:   .ASCII  \INSTALHLP\                          ;
                                               001D1          .BLKB   3                                    ;
                                               001D4 P.ACB:   .LONG   9                                    ;
                                               001D8          .ADDRESS P.ACC                               ;


                                                                .PSECT  $CODE$,NOWRT,2

                                         0000 00000             .ENTRY  INS$HELP_VERB, Save nothing          ; 0726
                              0000'   CF  9F 00002             PUSHAB  HELP_DESC                           ; 0737
                              0000'   CF  9F 00006             PUSHAB  P.ABZ                               ;
                00000000G     00        02 FB 0000A             CALLS   #2, CLI$GET_VALUE                   ;
                           00000000G    00 9F 00011             PUSHAB  LIB$GET_INPUT                       ; 0740
                                        7E D4 00017             CLRL    -(SP)                               ;
                              0000'   CF  9F 00019             PUSHAB  P.ACB                               ;
                              0000'   CF  9F 0001D             PUSHAB  HELP_DESC                           ;
                                        7E D4 00021             CLRL    -(SP)                               ;
                           00000000G    00 9F 00023             PUSHAB  LIB$PUT_OUTPUT                      ;
                00000000G     00        06 FB 00029             CALLS   #6, LBR$OUTPUT_HELP                 ;
                              09        50 E8 00030             BLBS    STATUS, 1$                          ;
                                        50 DD 00033             PUSHL   STATUS                              ;
                00000000G     00        01 FB 00035             CALLS   #1, LIB$SIGNAL                      ;
                              50        01 D0 0003C 1$:         MOVL    #1, R0                              ; 0741
                                        04 0003F                RET                                         ; 0742

; Routine Size:  64 bytes,    Routine Base:  $CODE$ + 0234

;  549          0743  1
```

```
;  551          0744  1 %SBTTL  'INS$EXIT_VERB';
;  552          0745  1 GLOBAL ROUTINE INS$EXIT_VERB =
;  553          0746  1 !+++
;  554          0747  1 !
;  555          0748  1 !   New command interface action routine for EXIT
;  556          0749  1 !
;  557          0750  1 !---
;  558          0751  2 BEGIN
;  559          0752  2 INTERACTIVE = FALSE;
;  560          0753  2 RETURN TRUE;
;  561          0754  1 END;                                    ! Routine INS$EXIT_VERB


                                        0000 00000          .ENTRY  INS$EXIT_VERB, Save nothing     ; 0745
                                0000'   CF  D4 00002        CLRL    INTERACTIVE                     ; 0752
                             50             01  D0 00006    MOVL    #1, R0                          ; 0753
                                            04 00009        RET                                     ; 0754

; Routine Size:  10 bytes,    Routine Base:  $CODE$ + 0274

;  562          0755  1
```

```
;  564          0756  1 %SBTTL  'INS$CREATE_VERB';
;  565          0757  1 GLOBAL ROUTINE INS$CREATE_VERB =
;  566          0758  1 !+++
;  567          0759  1 !
;  568          0760  1 !   New command interface action routine for CREATE
;  569          0761  1 !
;  570          0762  1 !---
;  571          0763  2 BEGIN
;  572          0764  2 LOCAL
;  573          0765  2     PURGE,
;  574          0766  2     CMK_ARGLST : VECTOR [1],    ! arguement list for change mode dispatcher
;  575          0767  2     OPEN_STATUS,
;  576          0768  2     STATUS;
;  577          0769  2
;  578          0770  2 INS$GL_CTLMSK [INS$V_PROCESS] = CLI$PRESENT( PROCESS_DSC );
;  579          0771  2 IF NOT .INS$GL_CTLMSK [INS$V_PROCESS]
;  580          0772  2 THEN
;  581          0773  3     BEGIN
;  582          0774  3     !
;  583          0775  3     !   Since INSTALL may be installed with CMKRNL privilege,
;  584          0776  3     !   check that user has CMKRNL privilege before allowing
;  585          0777  3     !   any operations other than /PROCESS operations.
;  586          0778  3     !
;  587          0779  3     CMK_ARGLST [0] = 0;
;  588          0780  3     STATUS = $CMKRNL (ROUTIN = INS$CHECK_PRIV, ARGLST = CMK_ARGLST);
;  589          0781  3     IF NOT .STATUS
;  590          0782  3     THEN
;  591          0783  4         BEGIN
;  592          0784  4         SIGNAL (SS$_NOCMKRNL);
;  593          0785  4         RETURN TRUE;
;  594          0786  3         END;
;  595          0787  3
;  596          0788  3     INS$GL_CTLMSK [INS$V_PRIV] = CLI$PRESENT( PRIVILEGED_DSC );
;  597          0789  3     IF .INS$GL_CTLMSK [INS$V_PRIV]
;  598          0790  3     THEN
;  599          0791  4         BEGIN
;  600          0792  4         LOCAL
;  601          0793  4             ALL_PRIVS;  ! boolean
;  602          0794  4
;  603          0795  4         ALL_PRIVS = TRUE;                                   ! turn on all privs by default
;  604          0796  4         CH$FILL (0, 8, INS$GQ_KFEPRIVS);                    ! initialize privilege quadword to zeros
;  605          0797  4         WHILE CLI$GET_VALUE (PRIVILEGED_DSC, PRIV_DSC ) DO
;  606          0798  5             BEGIN
;  607          0799  5             ALL_PRIVS = FALSE;                              ! turn off default
;  608          0800  5             SELECT PRV$SETPRIV (PRIV_DSC, INS$GQ_KFEPRIVS) OF   ! set the appropriate bit in the mask
;  609          0801  5                 SET
;  610          0802  5
;  611          0803  5                 [PRV$_INVNAM]:
;  612          0804  6                     BEGIN
;  613          0805  6                     SIGNAL (INS$_INVPRVNAM, 1, PRIV_DSC);
;  614          0806  6                     RETURN TRUE;
;  615          0807  5                     END;
;  616          0808  5
;  617          0809  5                 [PRV$_NOTUNQ]:
;  618          0810  6                     BEGIN
;  619          0811  6                     SIGNAL (INS$_PRVNOTUNQ, 1, PRIV_DSC);
;  620          0812  6                     RETURN TRUE;
```

```
;  621     0813  5                    END;
;  622     0814  5                TES;
;  623     0815  4            END;                            ! end while there are more privs
;  624     0816  4        IF .ALL_PRIVS                       ! No privs specified with /PRIV so
;  625     0817  4        THEN CH$FILL (-1, 8, INS$GQ_KFEPRIVS);          ! use default of all privs set
;  626     0818  3        END;                                ! /PRIV
;  627     0819  3
;  628     0820  2    END;                                    ! Not /PROCESS
;  629     0821  2
;  630     0822  2 INS$GL_CTLMSK [INS$V_FILSPC] = CLI$PRESENT( FILE_SPEC_DSC );
;  631     0823  2
;  632     0824  2    INS$GL_CTLMSK [INS$V_WRITABLE] = CLI$PRESENT( WRITE_DSC );  ! Get WRITE before we open file
;  633     0825  2
;  634     0826  2    !
;  635     0827  2    ! Get the file spec from the CLI
;  636     0828  2    !
;  637     0829  2    REPORT( CLI$GET_VALUE ( FILE_SPEC_DSC, INS$GQ_FILDSC) );
;  638     0830  2    OPEN_STATUS = INS$OPEN_FILE ();
;  639     0831  3    IF (NOT .OPEN_STATUS)
;  640     0832  2    THEN
;  641     0833  3        BEGIN
;  642     0834  3        INS$CLOSE_FILE ();
;  643     0835  3        RETURN TRUE;
;  644     0836  2        END;
;  645     0837  2
;  646     0838  2    IF .INS$GL_KFEADR EQL 0
;  647     0839  2    THEN
;  648     0840  3        BEGIN
;  649     0841  3        INS$GL_CTLMSK [INS$V_NOPURGE] = FALSE;
;  650     0842  3        PURGE = CLI$PRESENT( PURGE_DSC );
;  651     0843  3        IF .PURGE EQL CLI$_NEGATED THEN INS$GL_CTLMSK [INS$V_NOPURGE] = TRUE;
;  652     0844  3
;  653     0845  3        INS$GL_CTLMSK [INS$V_PROTECT] = CLI$PRESENT( PROTECT_DSC );
;  654     0846  3        INS$GL_CTLMSK [INS$V_OPEN] = CLI$PRESENT( OPEN_DSC );
;  655     0847  3        INS$GL_CTLMSK [INS$V_HDRRES] = CLI$PRESENT( HDRRES_DSC );
;  656     0848  3
;  657     0849  3        INS$GL_CTLMSK [INS$V_SHARED] = CLI$PRESENT( SHARED_DSC );
;  658     0850  3        INS$GL_CTLMSK [INS$V_ACCOUNT] = CLI$PRESENT( ACCOUNT_DSC );
;  659     0851  3        INS$GL_CTLMSK [INS$V_EXEONLY] = CLI$PRESENT( XONLY_DSC );
;  660     0852  3
;  661     0853  3        STATUS = INS$CREATE ();          ! Create it
;  662     0854  3        END
;  663     0855  2    ELSE
;  664     0856  2        STATUS = INS$_EXISTS;
;  665     0857  2
;  666     0858  2    IF NOT .STATUS
;  667     0859  2    THEN
;  668     0860  3        BEGIN
;  669     0861  3        IF .STATUS EQL INS$_INTRNLERR
;  670     0862  3        THEN
;  671     0863  3            SIGNAL (INS$_FAIL,2,CREATE_DSC,INS$GQ_KFERNS,
;  672     0864  3                    .STATUS,1,.INS$L_INTRNLERR)
;  673     0865  3        ELSE
;  674     0866  3            SIGNAL (INS$_FAIL,2,CREATE_DSC,INS$GQ_KFERNS,.STATUS);
;  675     0867  3        END
;  676     0868  2    ELSE
;  677     0869  3        BEGIN
```

```
;  678          0870  3          !
;  679          0871  3          !   See if it should be logged
;  680          0872  3          !
;  681          0873  3          IF CLI$PRESENT( LOG_DSC )
;  682          0874  3          THEN
;  683          0875  4              BEGIN
;  684          0876  4              INS$GL_CTLMSK [INS$V_FULL] = TRUE;
;  685          0877  4              INS$LIST (.INS$GL_KFEADR);
;  686          0878  4              INS$GLOBAL (.INS$GL_KFEADR);
;  687          0879  3              END;
;  688          0880  2          END;
;  689          0881  2
;  690          0882  2      INS$CLOSE_FILE ();
;  691          0883  2
;  692          0884  2 RETURN TRUE;
;  693          0885  1 END;                                    ! routine INS$CREATE_VERB


                                                        .EXTRN  SYS$CMKRNL

                                                        .ENTRY  INS$CREATE_VERB, Save R2,R3,R4,R5,R6,R7,R8,-; 0757
                                                                R9,R10,R11
                                5B 00000000G   00   9E 00002    MOVAB   LIB$SIGNAL, R11                      ;
                                5A      0000'  CF   9E 00009    MOVAB   PRIVILEGED_DSC, R10                  ;
                                59 00000000G   00   9E 0000E    MOVAB   CLI$PRESENT, R9                      ;
                                58      0000'  CF   9E 00015    MOVAB   INS$GL_CTLMSK, R8                    ;
                                5E             04   C2 0001A    SUBL2   #4, SP                               ;
                                       28      AA   9F 0001D    PUSHAB  PROCESS_DSC                          ; 0770
                                69             01   FB 00020    CALLS   #1, CLI$PRESENT                      ;
        68          01         01             50   F0 00023    INSV    R0, #1, #1, INS$GL_CTLMSK            ;
                    03         68             01   E1 00028    BBC     #1, INS$GL_CTLMSK, 1$                ; 0771
                                            0099   31 0002C    BRW     8$                                   ;
                                              6E   D4 0002F 1$:  CLRL    CMK_ARGLST                           ; 0779
                                              5E   DD 00031      PUSHL   SP                                   ; 0780
                                        0000V CF   9F 00033      PUSHAB  INS$CHECK_PRIV                       ;
                            00000000G   00         02   FB 00037    CALLS   #2, SYS$CMKRNL                       ;
                                        57         50   D0 0003E    MOVL    R0, STATUS                           ;
                                        0A         57   E8 00041    BLBS    STATUS, 2$                           ; 0781
                                        7E    2804 8F   3C 00044    MOVZWL  #10244, -(SP)                        ; 0784
                                        6B         01   FB 00049    CALLS   #1, LIB$SIGNAL                       ;
                                                   6C   11 0004C    BRB     6$                                   ; 0785
                                                   5A   DD 0004E 2$:  PUSHL   R10                                  ; 0788
                                        69         01   FB 00050    CALLS   #1, CLI$PRESENT                      ;
    01      A8          01             07         50   F0 00053    INSV    R0, #7, #1, INS$GL_CTLMSK+1          ;
                                          01   A8   95 00059    TSTB    INS$GL_CTLMSK+1                      ; 0789
                                                   6A   18 0005C    BGEQ    8$                                   ;
                                        56         01   D0 0005E    MOVL    #1, ALL_PRIVS                        ; 0795
            08          00             6E         00   2C 00061    MOVC5   #0, (SP), #0, #8, INS$GQ_KFEPRIVS    ; 0796
                                          14   A8      00066
                                        0000' CF   9F 00068 3$:  PUSHAB  PRIV_DSC                             ; 0797
                                                   5A   DD 0006C    PUSHL   R10                                  ;
                            00000000G   00         02   FB 0006E    CALLS   #2, CLI$GET_VALUE                    ;
                                        45         50   E9 00075    BLBC    R0, 7$                               ;
                                                   56   D4 00078    CLRL    ALL_PRIVS                            ; 0799
                                          14   A8   9F 0007A    PUSHAB  INS$GQ_KFEPRIVS                      ; 0800
                                        0000' CF   9F 0007D      PUSHAB  PRIV_DSC                             ;
```

```
                    00000000G  00              02  FB 00081          CALLS   #2, PRV$SETPRIV                                  ;
                               52              50  D0 00088          MOVL    R0, R2                                          ;
                    00000000G  8F              52  D1 0008B          CMPL    R2, #PRV$_INVNAM                                ; 0803
                               0E              12  00092          BNEQ    4$                                              ;
                                        0000'  CF  9F 00094          PUSHAB  PRIV_DSC                                        ; 0805
                                               01  DD 00098          PUSHL   #1                                              ;
                                   00000000G   8F  DD 0009A          PUSHL   #INS$_INVPRVNAM                                 ;
                                               15  11 000A0          BRB     5$                                              ;
                    00000000G  8F              52  D1 000A2  4$:     CMPL    R2, #PRV$_NOTUNQ                                ; 0809
                               BD              12  000A9          BNEQ    3$                                              ;
                                        C000'  CF  9F 000AB          PUSHAB  PRIV_DSC                                        ; 0811
                                               01  DD 000AF          PUSHL   #1                                              ;
                                   00000000G   8F  DD 000B1          PUSHL   #INS$_PRVNOTUNQ                                 ;
                                          6B       03  FB 000B7  5$:     CALLS   #3, LIB$SIGNAL                              ;
                                             011C  31  000BA  6$:     BRW     17$                                             ; 0812
                                          08       56  E9 000BD  7$:     BLBC    ALL_PRIVS, 8$                               ; 0816
                08        FF    8F              6E  00  2C 000C0          MOVC5   #0, (SP), #-1, #8, INS$GQ_KFEPRIVS          ; 0817
                                            14    A8      000C6
                                            84    AA  9F 000C8  8$:     PUSHAB  FILE_SPEC_DSC                               ; 0822
                                          69       01  FB 000CB          CALLS   #1, CLI$PRESENT                             ;
                68              01              00  50  F0 000CE          INSV    R0, #0, #1, INS$GL_CTLMSK                   ;
                                            6C    AA  9F 000D3          PUSHAB  WRITE_DSC                                   ; 0824
                                          69       01  FB 000D6          CALLS   #1, CLI$PRESENT                             ;
        02      A8              01              02  50  F0 000D9          INSV    R0, #2, #1, INS$GL_CTLMSK+2                 ;
                                            1C    A8  9F 000DF          PUSHAB  INS$GQ_FILDSC                               ; 0829
                                            84    AA  9F 000E2          PUSHAB  FILE_SPEC_DSC                               ;
                                   00000000G   00  02  FB 000E5          CALLS   #2, CLI$GET_VALUE                           ;
                                               05  50  E8 000EC          BLBS    STATUS, 9$                                  ;
                                               50  DD 000EF          PUSHL   STATUS                                      ;
                                          6B       01  FB 000F1          CALLS   #1, LIB$SIGNAL                              ;
                                   0000V   CF      00  FB 000F4  9$:     CALLS   #0, INS$OPEN_FILE                           ; 0830
                                          03       50  E8 000F9          BLBS    OPEN_STATUS, 10$                            ; 0831
                                             00D5  31  000FC          BRW     16$                                             ;
                                            10    A8  D5 000FF  10$:    TSTL    INS$GL_KFEADR                               ; 0838
                                          6D       12  00102          BNEQ    12$                                             ;
                                      02    A8          08  8A 00104          BICB2   #8, INS$GL_CTLMSK+2                     ; 0841
                                            7C    AA  9F 00108          PUSHAB  PURGE_DSC                                   ; 0842
                                          69       01  FB 0010B          CALLS   #1, CLI$PRESENT                             ;
                           00000000G    8F          50  D1 0010E          CMPL    PURGE, #CLI$_NEGATED                        ; 0843
                                          04       12  00115          BNEQ    11$                                             ;
                                      02    A8          08  88 00117          BISB2   #8, INS$GL_CTLMSK+2                     ;
                                            58    AA  9F 0011B  11$:    PUSHAB  PROTECT_DSC                                 ; 0845
                                          69       01  FB 0011E          CALLS   #1, CLI$PRESENT                             ;
        02      A8              01              00  50  F0 00121          INSV    R0, #0, #1, INS$GL_CTLMSK+2                 ;
                                            44    AA  9F 00127          PUSHAB  OPEN_DSC                                    ; 0846
                                          69       01  FB 0012A          CALLS   #1, CLI$PRESENT                             ;
        01      A8              01              05  50  F0 0012D          INSV    R0, #5, #1, INS$GL_CTLMSK+1                 ;
                                            18    AA  9F 00133          PUSHAB  HDRRES_DSC                                  ; 0847
                                          69       01  FB 00136          CALLS   #1, CLI$PRESENT                             ;
        01      A8              01              06  50  F0 00139          INSV    R0, #6, #1, INS$GL_CTLMSK+1                 ;
                                            38    AA  9F 0013F          PUSHAB  SHARED_DSC                                  ; 0849
                                          69       01  FB 00142          CALLS   #1, CLI$PRESENT                             ;
        02      A8              01              01  50  F0 00145          INSV    R0, #1, #1, INS$GL_CTLMSK+2                 ;
                                          0090  CA  9F 0014B          PUSHAB  ACCOUNT_DSC                                 ; 0850
                                          69       01  FB 0014F          CALLS   #1, CLI$PRESENT                             ;
        02      A8              01              04  50  F0 00152          INSV    R0, #4, #1, INS$GL_CTLMSK+2                 ;
                                          00A4  CA  9F 00158          PUSHAB  XONLY_DSC                                   ; 0851
```

```
                                       69           01  FB 0015C           CALLS   #1, CLI$PRESENT                    ;
      02   A8              01          05           50  F0 0015F           INSV    R0, #5, #1, INS$GL_CTLMSK+2        ;
                            00000000G  00           00  FB 00165           CALLS   #0, INS$CREATE                     ; 0853
                                       57           50  D0 0016C           MOVL    R0, STATUS                         ;
                                                    07  11 0016F           BRB     13$                                ; 0838
                                       57 00000000G 8F  D0 00171 12$:      MOVL    #INS$_EXISTS, STATUS               ; 0856
                                       38           57  E8 00178 13$:      BLBS    STATUS, 15$                        ; 0858
                            00000000G  8F           57  D1 0017B           CMPL    STATUS, #INS$_INTRNLERR            ; 0861
                                                    1A  12 00182           BNEQ    14$                                ;
                                                98  A8  DD 00184           PUSHL   INS$L_INTRNLERR                    ; 0864
                                                    01  DD 00187           PUSHL   #1                                 ; 0863
                                                    57  DD 00189           PUSHL   STATUS                             ; 0864
                                                08  A8  9F 0018B           PUSHAB  INS$GQ_KFERNS                      ; 0863
                                                94  AA  9F 0018E           PUSHAB  CREATE_DSC                         ;
                                                    02  DD 00191           PUSHL   #2                                 ;
                                          00000000G 8F  DD 00193           PUSHL   #INS$_FAIL                         ;
                                       6B           07  FB 00199           CALLS   #7, LIB$SIGNAL                     ;
                                                    36  11 0019C           BRB     16$                                ;
                                                    57  DD 0019E 14$:      PUSHL   STATUS                             ; 0866
                                                08  A8  9F 001A0           PUSHAB  INS$GQ_KFERNS                      ;
                                                94  AA  9F 001A3           PUSHAB  CREATE_DSC                         ;
                                                    02  DD 001A6           PUSHL   #2                                 ;
                                          00000000G 8F  DD 001A8           PUSHL   #INS$_FAIL                         ;
                                       6B           05  FB 001AE           CALLS   #5, LIB$SIGNAL                     ;
                                                    21  11 001B1           BRB     16$                                ; 0858
                                                DC  AA  9F 001B3 15$:      PUSHAB  LOG_DSC                            ; 0873
                                       69           01  FB 001B6           CALLS   #1, CLI$PRESENT                    ;
                                       18           50  E9 001B9           BLBC    R0, 16$                            ;
                                  01   A8           04  88 001BC           BISB2   #4, INS$GL_CTLMSK+1                ; 0876
                                                10  A8  DD 001C0           PUSHL   INS$GL_KFEADR                      ; 0877
                            00000000G  00           01  FB 001C3           CALLS   #1, INS$LIST                       ;
                                                10  A8  DD 001CA           PUSHL   INS$GL_KFEADR                      ; 0878
                            00000000G  00           01  FB 001CD           CALLS   #1, INS$GLOBAL                     ;
                                0000V  CF           00  FB 001D4 16$:      CALLS   #0, INS$CLOSE_FILE                 ; 0882
                                       50           01  D0 001D9 17$:      MOVL    #1, R0                             ; 0884
                                                        04 001DC           RET                                        ; 0885

; Routine Size:  477 bytes,    Routine Base:  $CODE$ + 027E


;  694         0886  1
```

```
;  696          0887  1 %SBTTL  'INS$REPLACE_VERB';
;  697          0888  1 GLOBAL ROUTINE INS$REPLACE_VERB =
;  698          0889  1 !+++
;  699          0890  1 !
;  700          0891  1 !   New command interface action routine for REPLACE
;  701          0892  1 !
;  702          0893  1 !---
;  703          0894  2 BEGIN
;  704          0895  2 LOCAL
;  705          0896  2     DELETED,
;  706          0897  2     QUAL_PRESENT,
;  707          0898  2     CMK_ARGLST : VECTOR [1],    ! arguement list for change mode dispatcher
;  708          0899  2     OPEN_STATUS,
;  709          0900  2     STATUS;
;  710          0901  2
;  711          0902  2 INS$GL_CTLMSK [INS$V_PROCESS] = CLI$PRESENT( PROCESS_DSC );
;  712          0903  2 IF NOT .INS$GL_CTLMSK [INS$V_PROCESS]
;  713          0904  2 THEN
;  714          0905  3     BEGIN
;  715          0906  3     !
;  716          0907  3     !   Since INSTALL may be installed with CMKRNL privilege,
;  717          0908  3     !   check that user has CMKRNL privilege before allowing
;  718          0909  3     !   any operations other than /PROCESS operations.
;  719          0910  3     !
;  720          0911  3     CMK_ARGLST [0] = 0;
;  721          0912  3     STATUS = $CMKRNL (ROUTIN = INS$CHECK_PRIV, ARGLST = CMK_ARGLST);
;  722          0913  3     IF NOT .STATUS
;  723          0914  3     THEN
;  724          0915  4         BEGIN
;  725          0916  4         SIGNAL (SS$_NOCMKRNL);
;  726          0917  4         RETURN TRUE;
;  727          0918  3         END;
;  728          0919  2     END;
;  729          0920  2
;  730          0921  2 INS$GL_CTLMSK [INS$V_FILSPC] = CLI$PRESENT( FILE_SPEC_DSC );
;  731          0922  2
;  732          0923  2     !
;  733          0924  2     ! Get the file spec from the CLI
;  734          0925  2     !
;  735          0926  2     REPORT( CLI$GET_VALUE ( FILE_SPEC_DSC, INS$GQ_FILDSC) );
;  736          0927  2     OPEN_STATUS = INS$OPEN_FILE ();
;  737          0928  3     IF (NOT .OPEN_STATUS)
;  738          0929  2     THEN
;  739          0930  3         BEGIN
;  740          0931  3         INS$CLOSE_FILE ();
;  741          0932  3         RETURN TRUE;
;  742          0933  2         END;
;  743          0934  2
;  744          0935  2     !
;  745          0936  2     !   Delete and recreate with new attributes
;  746          0937  2     !
;  747          0938  2     INS$GL_CTLMSK [INS$V_DELETE] = TRUE;        ! set delete flag
;  748          0939  2     INS$GL_CTLMSK [INS$V_REPLACE] = TRUE;       ! set replace flag so that attributes
;  749          0940  2                                                 ! will be returned in INS$GL_REPLACE_MSK
;  750          0941  2     INS$GL_REPLACE_MSK = 0;
;  751          0942  2     STATUS = INS$DELETE ();                     ! recreate it
;  752          0943  2     IF NOT .STATUS
```

```
;  753          0944  2      THEN
;  754          0945  3          BEGIN
;  755          0946  3          DELETED = FALSE;                        ! Record status of delete step
;  756          0947  3
;  757          0948  3          IF .STATUS EQL INS$_INTRNLERR
;  758          0949  3          THEN
;  759          0950  3              SIGNAL (((INS$_FAIL AND NOT STS$M_SEVERITY) OR STS$K_WARNING),
;  760          0951  3                      2,DELETE_DSC,INS$GQ_KFERNS,
;  761          0952  3                      .STATUS,1,.INS$L_INTRNLERR);
;  762          0953  3          END
;  763          0954  2      ELSE
;  764          0955  2          DELETED = TRUE;                         ! Record status of delete step
;  765          0956  2
;  766          0957  2      INS$GL_KFEADR = 0;                          ! Entry has gone away
;  767          0958  2      INS$CLOSE_FILE ();
;  768          0959  2
;  769          0960  2      INS$GL_CTLMSK = .INS$GL_REPLACE_MSK;        ! use replace mask
;  770          0961  2      INS$GL_CTLMSK [INS$V_CREATE] = TRUE;        ! set create flag
;  771          0962  2      INS$GL_CTLMSK [INS$V_DELETE] = FALSE;       ! clear delete flag
;  772          0963  2
;  773          0964  2      !
;  774          0965  2      !   If any qualifiers were specified with REPLACE then
;  775          0966  2      !   check they override the current installed attributes.
;  776          0967  2      !   If they are negated they serve to remove the current attribute.
;  777          0968  2      !
;  778          0969  2      QUAL_PRESENT = CLI$PRESENT( PURGE_DSC );
;  779          0970  2      IF .QUAL_PRESENT THEN INS$GL_CTLMSK [INS$V_NOPURGE] = FALSE;
;  780          0971  2      IF .QUAL_PRESENT EQL CLI$_NEGATED THEN INS$GL_CTLMSK [INS$V_NOPURGE] = TRUE;
;  781          0972  2
;  782          0973  2      QUAL_PRESENT = CLI$PRESENT( PROTECT_DSC );
;  783          0974  2      IF .QUAL_PRESENT THEN INS$GL_CTLMSK [INS$V_PROTECT] = TRUE;
;  784          0975  2      IF .QUAL_PRESENT EQL CLI$_NEGATED THEN INS$GL_CTLMSK [INS$V_PROTECT] = FALSE;
;  785          0976  2
;  786          0977  2      QUAL_PRESENT = CLI$PRESENT( OPEN_DSC );
;  787          0978  2      IF .QUAL_PRESENT THEN INS$GL_CTLMSK [INS$V_OPEN] = TRUE;
;  788          0979  2      IF .QUAL_PRESENT EQL CLI$_NEGATED THEN INS$GL_CTLMSK [INS$V_OPEN] = FALSE;
;  789          0980  2
;  790          0981  2      QUAL_PRESENT = CLI$PRESENT( HDRRES_DSC );
;  791          0982  2      IF .QUAL_PRESENT THEN INS$GL_CTLMSK [INS$V_HDRRES] = TRUE;
;  792          0983  2      IF .QUAL_PRESENT EQL CLI$_NEGATED THEN INS$GL_CTLMSK [INS$V_HDRRES] = FALSE;
;  793          0984  2
;  794          0985  2      QUAL_PRESENT = CLI$PRESENT( PRIVILEGED_DSC );
;  795          0986  2      IF .QUAL_PRESENT
;  796          0987  2      THEN
;  797          0988  3          BEGIN
;  798          0989  3          LOCAL
;  799          0990  3              ALL_PRIVS;  ! boolean
;  800          0991  3
;  801          0992  3          INS$GL_CTLMSK [INS$V_PRIV] = TRUE;
;  802          0993  3          ALL_PRIVS = TRUE;                                   ! turn on all privs by default
;  803          0994  3          CH$FILL (0, 8, INS$GQ_KFEPRIVS);                    ! initialize privilege quadword to zeros
;  804          0995  3          WHILE CLI$GET_VALUE (PRIVILEGED_DSC, PRIV_DSC ) DO
;  805          0996  4              BEGIN
;  806          0997  4              ALL_PRIVS = FALSE;                              ! turn off default
;  807          0998  4              SELECT PRV$SETPRIV (PRIV_DSC, INS$GQ_KFEPRIVS) OF   ! set the appropriate bit in the mask
;  808          0999  4                  SET
;  809          1000  4
```

```
;  810          1001  4                  [PRV$_INVNAM]:
;  811          1002  5                      BEGIN
;  812          1003  5                      SIGNAL (INS$_INVPRVNAM, 1, PRIV_DSC);
;  813          1004  5                      RETURN TRUE;
;  814          1005  4                      END;
;  815          1006  4
;  816          1007  4                  [PRV$_NOTUNQ]:
;  817          1008  5                      BEGIN
;  818          1009  5                      SIGNAL (INS$_PRVNOTUNQ, 1, PRIV_DSC);
;  819          1010  5                      RETURN TRUE;
;  820          1011  4                      END;
;  821          1012  4                  TES;
;  822          1013  3              END;                          ! end while there are more privs
;  823          1014  3          IF .ALL_PRIVS                     ! No privs specified with /PRIV so
;  824          1015  3          THEN CH$FILL (-1, 8, INS$GQ_KFEPRIVS);    ! use default of all privs set
;  825          1016  2          END;                              ! /PRIV
;  826          1017  2      IF .QUAL_PRESENT EQL CLI$_NEGATED THEN INS$GL_CTLMSK [INS$V_PRIV] = FALSE;
;  827          1018  2
;  828          1019  2      QUAL_PRESENT = CLI$PRESENT( SHARED_DSC );
;  829          1020  2      IF .QUAL_PRESENT THEN INS$GL_CTLMSK [INS$V_SHARED] = TRUE;
;  830          1021  2      IF .QUAL_PRESENT EQL CLI$_NEGATED THEN INS$GL_CTLMSK [INS$V_SHARED] = FALSE;
;  831          1022  2
;  832          1023  2      IF .INS$GL_CTLMSK [INS$V_SHARED]
;  833          1024  2      THEN
;  834          1025  3          BEGIN
;  835          1026  3          QUAL_PRESENT = CLI$PRESENT( WRITE_DSC );
;  836          1027  3          IF .QUAL_PRESENT THEN INS$GL_CTLMSK [INS$V_WRITABLE] = TRUE;
;  837          1028  3          IF .QUAL_PRESENT EQL CLI$_NEGATED THEN INS$GL_CTLMSK [INS$V_WRITABLE] = FALSE;
;  838          1029  3          END
;  839          1030  2      ELSE
;  840          1031  2          INS$GL_CTLMSK [INS$V_WRITABLE] = FALSE;
;  841          1032  2
;  842          1033  2      QUAL_PRESENT = CLI$PRESENT( ACCOUNT_DSC );
;  843          1034  2      IF .QUAL_PRESENT THEN INS$GL_CTLMSK [INS$V_ACCOUNT] = TRUE;
;  844          1035  2      IF .QUAL_PRESENT EQL CLI$_NEGATED THEN INS$GL_CTLMSK [INS$V_ACCOUNT] = FALSE;
;  845          1036  2
;  846          1037  2      QUAL_PRESENT = CLI$PRESENT( XONLY_DSC );
;  847          1038  2      IF .QUAL_PRESENT THEN INS$GL_CTLMSK [INS$V_EXEONLY] = TRUE;
;  848          1039  2      IF .QUAL_PRESENT EQL CLI$_NEGATED THEN INS$GL_CTLMSK [INS$V_EXEONLY] = FALSE;
;  849          1040  2
;  850          1041  2      INS$OPEN_FILE ();
;  851          1042  2
;  852          1043  2      IF .INS$GL_KFEADR EQL 0
;  853          1044  2      THEN
;  854          1045  2          STATUS = INS$CREATE ()          ! recreate it
;  855          1046  2      ELSE
;  856          1047  2          STATUS = INS$_EXISTS;
;  857          1048  2
;  858          1049  2      IF NOT .STATUS
;  859          1050  2      THEN
;  860          1051  3          BEGIN
;  861          1052  3          IF .STATUS EQL INS$_INTRNLERR
;  862          1053  3          THEN
;  863          1054  3              SIGNAL (INS$_FAIL,2,REPLACE_DSC,INS$GQ_KFERNS,
;  864          1055  3                      .STATUS,1,.INS$[_INTRNLERR)
;  865          1056  3          ELSE
;  866          1057  4              BEGIN
```

```
;  867          1058  4              SIGNAL (INS$_FAIL,2,REPLACE_DSC,INS$GQ_KFERNS,.STATUS);
;  868          1059  4              IF .DELETED
;  869          1060  4              THEN
;  870          1061  4                  SIGNAL (INS$_DELETED);
;  871          1062  3              END;
;  872          1063  3          END
;  873          1064  2      ELSE
;  874          1065  3          BEGIN
;  875          1066  3          IF NOT .DELETED
;  876          1067  3          THEN
;  877          1068  3              SIGNAL (INS$_NOPREV,1,INS$GQ_KFERNS); ! Report qualified success
;  878          1069  3
;  879          1070  3          !
;  880          1071  3          !   See if it should be logged
;  881          1072  3          !
;  882          1073  3          IF CLI$PRESENT( LOG_DSC )
;  883          1074  3          THEN
;  884          1075  4              BEGIN
;  885          1076  4              INS$GL_CTLMSK [INS$V_FULL] = TRUE;
;  886          1077  4              INS$LIST (.INS$GL_KFEADR);
;  887          1078  4              INS$GLOBAL (.INS$GL_KFEADR);
;  888          1079  3              END;
;  889          1080  2          END;
;  890          1081  2
;  891          1082  2
;  892          1083  2 RETURN TRUE;
;  893          1084  1 END;                                    ! routine INS$REPLACE_VERB


                                        0FFC 00000          .ENTRY  INS$REPLACE_VERB, Save R2,R3,R4,R5,R6,R7,-      ; 0888
                                                                    R8,R9,R10,R11                                       ;
                                5B 00000000G    00  9E 00002        MOVAB   CLI$PRESENT, R11                        ;
                                5A      0000'   CF  9E 00009        MOVAB   INS$GL_CTLMSK, R10                      ;
                                5E              04  C2 0000E        SUBL2   #4, SP                                  ;
                                        0000'   CF  9F 00011        PUSHAB  PROCESS_DSC                             ; 0902
                                6B              01  FB 00015        CALLS   #1, CLI$PRESENT                         ;
6A              01              01              50  F0 00018        INSV    R0, #1, #1, INS$GL_CTLMSK               ;
                1D              6A              01  E0 0001D        BBS     #1, INS$GL_CTLMSK, 1$                   ; 0903
                                6E              D4  00021        CLRL    CMK_ARGLST                              ; 0911
                                5E              DD  00023        PUSHL   SP                                      ; 0912
                                        0000V   CF  9F 00025        PUSHAB  INS$CHECK_PRIV                          ;
                    00000000G   00              02  FB 00029        CALLS   #2, SYS$CMKRNL                          ;
                                58              50  D0 00030        MOVL    R0, STATUS                              ;
                                08              58  E8 00033        BLBS    STATUS, 1$                              ; 0913
                                7E      2804    8F  3C 00036        MOVZWL  #10244, -(SP)                           ; 0916
                                        027A    31  03B              BRW     32$                                     ;
                                        0000'   CF  9F 0003E 1$:     PUSHAB  FILE_SPEC_DSC                           ; 0921
                                6B              01  FB 00042        CALLS   #1, CLI$PRESENT                         ;
6A              01              00              50  F0 00045        INSV    R0, #0, #1, INS$GL_CTLMSK               ;
                                        1C      AA  9F 0004A        PUSHAB  INS$GQ_FILDSC                           ; 0926
                                        0000'   CF  9F 0004D        PUSHAB  FILE_SPEC_DSC                           ;
                    00000000G   00              02  FB 00051        CALLS   #2, CLI$GET_VALUE                       ;
                                09              50  E8 00058        BLBS    STATUS, 2$                              ;
                                                50  DD 0005B        PUSHL   STATUS                                  ;
```

```
                00000000G  00              01  FB 0005D                CALLS   #1, LIB$SIGNAL                          ;
                    0000V  CF              00  FB 00064 2$:            CALLS   #0, INS$OPEN_FILE                       ; 0927
                           08              50  E8 00069                BLBS    OPEN_STATUS, 3$                         ; 0928
                    0000V  CF              00  FB 0006C                CALLS   #0, INS$CLOSE_FILE                      ; 0931
                                     0284  31  00071                BRW     35$                                     ; 0932
                           6A      60      8F  88 00074 3$:            BISB2   #96, INS$GL_CTLMSK                      ; 0939
                                   04      AA  D4 00078                CLRL    INS$GL_REPLACE_MSK                      ; 0941
                00000000G  00              00  FB 0007B                CALLS   #0, INS$DELETE                          ; 0942
                           58              50  D0 00082                MOVL    R0, STATUS                              ;
                           2A              58  E8 00085                BLBS    STATUS, 4$                              ; 0943
                                           59  D4 00088                CLRL    DELETED                                 ; 0946
                00000000G  8F              58  D1 0008A                CMPL    STATUS, #INS$_INTRNLERR                 ; 0948
                                           22  12 00091                BNEQ    5$                                      ;
                                     98    AA  DD 00093                PUSHL   INS$L_INTRNLERR                         ; 0952
                                           01  DD 00096                PUSHL   #1                                      ; 0950
                                           58  DD 00098                PUSHL   STATUS                                  ; 0952
                                     08    AA  9F 0009A                PUSHAB  INS$GQ_KFERNS                           ; 0950
                                   0000'   CF  9F 0009D                PUSHAB  DELETE_DSC                              ;
                                           02  DD 000A1                PUSHL   #2                                      ;
                                00000000*  8F  DD 000A3                PUSHL   #<INS$_FAIL&-8>                         ;
                00000000G  00              07  FB 000A9                CALLS   #7, LIB$SIGNAL                          ;
                                           03  11 000B0                BRB     5$                                      ; 0943
                           59              01  D0 000B2 4$:            MOVL    #1, DELETED                             ; 0955
                                     10    AA  D4 000B5 5$:            CLRL    INS$GL_KFEADR                           ; 0957
                    0000V  CF              00  FB 000B8                CALLS   #0, INS$CLOSE_FILE                      ; 0958
                           6A      04      AA  D0 000BD                MOVL    INS$GL_REPLACE_MSK, INS$GL_CTLMSK        ; 0960
                           6A      10      88  8A 000C1                BISB2   #16, INS$GL_CTLMSK                      ; 0961
                           6A      40      8F  8A 000C4                BICB2   #64, INS$GL_CTLMSK                      ; 0962
                                   0000'   CF  9F 000C8                PUSHAB  PURGE_DSC                               ; 0969
                           6B              01  FB 000CC                CALLS   #1, CLI$PRESENT                         ;
                           56              50  D0 000CF                MOVL    R0, QUAL_PRESENT                        ;
                           04              56  E9 000D2                BLBC    QUAL_PRESENT, 6$                        ; 0970
                        02 AA              08  8A 000D5                BICB2   #8, INS$GL_CTLMSK+2                     ;
                00000000G  8F              56  D1 000D9 6$:            CMPL    QUAL_PRESENT, #CLI$_NEGATED             ; 0971
                                           04  12 000E0                BNEQ    7$                                      ;
                        02 AA              08  88 000E2                BISB2   #8, INS$GL_CTLMSK+2                     ;
                                   0000'   CF  9F 000E6 7$:            PUSHAB  PROTECT_DSC                             ; 0973
                           6B              01  FB 000EA                CALLS   #1, CLI$PRESENT                         ;
                           56              50  D0 000ED                MOVL    R0, QUAL_PRESENT                        ;
                           04              56  E9 000F0                BLBC    QUAL_PRESENT, 8$                        ; 0974
                        02 AA              01  88 000F3                BISB2   #1, INS$GL_CTLMSK+2                     ;
                00000000G  8F              56  D1 000F7 8$:            CMPL    QUAL_PRESENT, #CLI$_NEGATED             ; 0975
                                           04  12 000FE                BNEQ    9$                                      ;
                        02 AA              01  8A 00100                BICB2   #1, INS$GL_CTLMSK+2                     ;
                                   0000'   CF  9F 00104 9$:            PUSHAB  OPEN_DSC                                ; 0977
                           6B              01  FB 00108                CALLS   #1, CLI$PRESENT                         ;
                           56              50  D0 0010B                MOVL    R0, QUAL_PRESENT                        ;
                           04              56  E9 0010E                BLBC    QUAL_PRESENT, 10$                       ; 0978
                        01 AA              20  88 00111                BISB2   #32, INS$GL_CTLMSK+1                    ;
                00000000G  8F              56  D1 00115 10$:           CMPL    QUAL_PRESENT, #CLI$_NEGATED             ; 0979
                                           04  12 0011C                BNEQ    11$                                     ;
                        01 AA              20  8A 0011E                BICB2   #32, INS$GL_CTLMSK+1                    ;
                                   0000'   CF  9F 00122 11$:           PUSHAB  HDRRES_DSC                              ; 0981
                           6B              01  FB 00126                CALLS   #1, CLI$PRESENT                         ;
                           56              50  D0 00129                MOVL    R0, QUAL_PRESENT                        ;
                           05              56  E9 0012C                BLBC    QUAL_PRESENT, 12$                       ; 0982
                        01 AA      40      8F  88 0012F                BISB2   #64, INS$GL_CTLMSK+1                    ;
```

```
                    00000000G  8F              56  D1 00134 12$:   CMPL    QUAL_PRESENT, #CLI$_NEGATED             ; 0983
                                                05  12 00138       BNEQ    13$
                         01    AA      40       8F  8A 0013D       BICB2   #64, INS$GL_CTLMSK+1
                               0000'   CF       9F  00142 13$:     PUSHAB  PRIVILEGED_DSC                          ; 0985
                                       6B       01  FB 00146       CALLS   #1, CLI$PRESENT
                                       56       50  D0 00149       MOVL    R0, QUAL_PRESENT
                                       75       56  E9 0014C       BLBC    QUAL_PRESENT, 18$                       ; 0986
                         01    AA      80       8F  88 0014F       BISB2   #128, INS$GL_CTLMSK+1                   ; 0992
                                       57       01  D0 00154       MOVL    #1, ALL_PRIVS                           ; 0993
                08       00            6E       00  2C 00157       MOVC5   #0, (SP), #0, #8, INS$GQ_KFEPRIVS       ; 0994
                                                14  AA   0015C
                               0000'   CF       9F  0015E 14$:     PUSHAB  PRIV_DSC                                ; 0995
                               0000'   CF       9F  00162          PUSHAB  PRIVILEGED_DSC
                    00000000G  00               02  FB 00166       CALLS   #2, CLI$GET_VALUE
                                       49       50  E9 0016D       BLBC    R0, 17$
                                                57  D4 00170       CLRL    ALL_PRIVS                               ; 0997
                                            14  AA  9F 00172       PUSHAB  INS$GQ_KFEPRIVS                         ; 0998
                               0000'   CF       9F  00175          PUSHAB  PRIV_DSC
                    00000000G  00               02  FB 00179       CALLS   #2, PRV$SETPRIV
                                       52       50  D0 00180       MOVL    R0, R2
                    00000000G  8F               52  D1 00183       CMPL    R2, #PRV$_INVNAM                        ; 1001
                                                0E  12 0018A       BNEQ    15$
                               0000'   CF       9F  0018C          PUSHAB  PRIV_DSC                                ; 1003
                                       01       DD  00190          PUSHL   #1
                          00000000G    8F       DD  00192          PUSHL   #INS$_INVPRVNAM
                                                15  11 00198       BRB     16$
                    00000000G  8F               52  D1 0019A 15$:  CMPL    R2, #PRV$_NOTUNQ                        ; 1007
                                                BB  12 001A1       BNEQ    14$
                               0000'   CF       9F  001A3          PUSHAB  PRIV_DSC                                ; 1009
                                       01       DD  001A7          PUSHL   #1
                          00000000G    8F       DD  001A9          PUSHL   #INS$_PRVNOTUNQ
                    00000000G  00               03  FB 001AF 16$:  CALLS   #3, LIB$SIGNAL
                                             013F   31  001B6       BRW     35$                                     ; 1010
                                       08       57  E9 001B9 17$:  BLBC    ALL_PRIVS, 18$                          ; 1014
                08       FF    8F       6E       00  2C 001BC       MOVC5   #0, (SP), #-1, #8, INS$GQ_KFEPRIVS      ; 1015
                                                14  AA   001C2
                    00000000G  8F               56  D1 001C4 18$:  CMPL    QUAL_PRESENT, #CLI$_NEGATED             ; 1017
                                                05  12 001CB       BNEQ    19$
                         01    AA      80       8F  8A 001CD       BICB2   #128, INS$GL_CTLMSK+1
                               0000'   CF       9F  001D2 19$:     PUSHAB  SHARED_DSC                              ; 1019
                                       6B       01  FB 001D6       CALLS   #1, CLI$PRESENT
                                       56       50  D0 001D9       MOVL    R0, QUAL_PRESENT
                                       04       56  E9 001DC       BLBC    QUAL_PRESENT, 20$                       ; 1020
                         02    AA               02  88 001DF       BISB2   #2, INS$GL_CTLMSK+2
                    00000000G  8F               56  D1 001E3 20$:  CMPL    QUAL_PRESENT, #CLI$_NEGATED             ; 1021
                                                04  12 001EA       BNEQ    21$
                         02    AA               02  8A 001EC       BICB2   #2, INS$GL_CTLMSK+2
                    1A   02    AA               01  E1 001F0 21$:  BBC     #1, INS$GL_CTLMSK+2, 23$                ; 1023
                               0000'   CF       9F  001F5          PUSHAB  WRITE_DSC                               ; 1026
                                       6B       01  FB 001F9       CALLS   #1, CLI$PRESENT
                                       56       50  D0 001FC       MOVL    R0, QUAL_PRESENT
                                       04       56  E9 001FF       BLBC    QUAL_PRESENT, 22$                       ; 1027
                         02    AA               04  88 00202       BISB2   #4, INS$GL_CTLMSK+2
                    00000000G  8F               56  D1 00206 22$:  CMPL    QUAL_PRESENT, #CLI$_NEGATED             ; 1028
                                                04  12 0020D       BNEQ    24$
                         02    AA               04  8A 0020F 23$:  BICB2   #4, INS$GL_CTLMSK+2                    ; 1031
                               0000'   CF       9F  00213 24$:     PUSHAB  ACCOUNT_DSC                             ; 1033
```

```
                    6B              01 FB 00217          CALLS   #1, CLI$PRESENT                          ;
                    56              50 D0 0021A          MOVL    R0, QUAL_PRESENT                         ;
                    04              56 E9 0021D          BLBC    QUAL_PRESENT, 25$                        ; 1034
                 02 AA              10 88 00220          BISB2   #16, INS$GL_CTLMSK+2                     ;
00000000G        8F                 56 D1 00224 25$:     CMPL    QUAL_PRESENT, #CLI$_NEGATED              ; 1035
                                    04 12 0022B          BNEQ    26$                                      ;
                 02 AA              10 8A 0022D          BICB2   #16, INS$GL_CTLMSK+2                     ;
                          0000'     CF 9F 00231 26$:     PUSHAB  XONLY_DSC                                ; 1037
                    6B              01 FB 00235          CALLS   #1, CLI$PRESENT                          ;
                    56              50 D0 00238          MOVL    R0, QUAL_PRESENT                         ;
                    04              56 E9 0023B          BLBC    QUAL_PRESENT, 27$                        ; 1038
                 02 AA              20 88 0023E          BISB2   #32, INS$GL_CTLMSK+2                     ;
00000000G        8F                 56 D1 00242 27$:     CMPL    QUAL_PRESENT, #CLI$_NEGATED              ; 1039
                                    04 12 00249          BNEQ    28$                                      ;
                 02 AA              20 8A 0024B          BICB2   #32, INS$GL_CTLMSK+2                     ;
              0000V CF              00 FB 0024F 28$:     CALLS   #0, INS$OPEN_FILE                        ; 1041
                          10    AA  D5 00254             TSTL    INS$GL_KFEADR                            ; 1043
                                0C  12 00257             BNEQ    29$                                      ;
00000000G        00                 00 FB 00259          CALLS   #0, INS$CREATE                           ; 1045
                    58              50 D0 00260          MOVL    R0, STATUS                               ;
                                    07 11 00263          BRB     30$                                      ;
                    58 00000000G    8F D0 00265 29$:     MOVL    #INS$_EXISTS, STATUS                     ; 1047
                    52              58 E8 0026C 30$:     BLBS    STATUS, 33$                              ; 1049
00000000G        8F                 58 D1 0026F          CMPL    STATUS, #INS$_INTRNLERR                  ; 1052
                                    1F 12 00276          BNEQ    31$                                      ;
                          98    AA  DD 00278             PUSHL   INS$L_INTRNLERR                          ; 1055
                                01  DD 0027B             PUSHL   #1                                       ; 1054
                                58  DD 0027D             PUSHL   STATUS                                   ; 1055
                             08 AA  9F 0027F             PUSHAB  INS$GQ_KFERNS                            ; 1054
                          0000' CF  9F 00282             PUSHAB  REPLACE_DSC                              ;
                                02  DD 00286             PUSHL   #2                                       ;
                 00000000G      8F  DD 00288             PUSHL   #INS$_FAIL                               ;
00000000G        00                 07 FB 0028E          CALLS   #7, LIB$SIGNAL                           ;
                                    61 11 00295          BRB     35$                                      ;
                                    58 DD 00297 31$:     PUSHL   STATUS                                   ; 1058
                             08 AA  9F 00299             PUSHAB  INS$GQ_KFERNS                            ;
                          0000' CF  9F 0029C             PUSHAB  REPLACE_DSC                              ;
                                02  DD 002A0             PUSHL   #2                                       ;
                 00000000G      8F  DD 002A2             PUSHL   #INS$_FAIL                               ;
00000000G        00                 05 FB 002A8          CALLS   #5, LIB$SIGNAL                           ;
                    46              59 E9 002AF          BLBC    DELETED, 35$                             ; 1059
                 00000000G      8F  DD 002B2             PUSHL   #INS$_DELETED                            ; 1061
00000000G        00                 01 FB 002B8 32$:     CALLS   #1, LIB$SIGNAL                           ;
                                    37 11 002BF          BRB     35$                                      ; 1049
                    12              59 E8 002C1 33$:     BLBS    DELETED, 34$                             ; 1066
                             08 AA  9F 002C4             PUSHAB  INS$GQ_KFERNS                            ; 1068
                                01  DD 002C7             PUSHL   #1                                       ;
                 00000000G      8F  DD 002C9             PUSHL   #INS$_NOPREV                             ;
00000000G        00                 03 FB 002CF          CALLS   #3, LIB$SIGNAL                           ;
                          0000' CF  9F 002D6 34$:     PUSHAB  _OG_DSC                                  ; 1073
                    6B              01 FB 002DA          CALLS   #1, CLI$PRESENT                          ;
                    18              50 E9 002DD          BLBC    R0, 35$                                  ;
                 01 AA              04 88 002E0          BISB2   #4, INS$GL_CTLMSK+1                      ; 1076
                          10    AA  DD 002E4             PUSHL   INS$GL_KFEADR                            ; 1077
00000000G        00                 01 FB 002E7          CALLS   #1, INS$LIST                             ;
                          10    AA  DD 002EE             PUSHL   INS$GL_KFEADR                            ; 1078
00000000G        00                 01 FB 002F1          CALLS   #1, INS$GLOBAL                           ;
```

```
                                50              01  D0 002F8 35$:    MOVL    #1, R0                          ; 1083
                                                04  002FB           RET                                     ; 1084

; Routine Size:  764 bytes,    Routine Base:  $CODE$ + 045B
```

```
;  895          1085  1 %SBTTL  'INS$REMOVE_VERB';
;  896          1086  1 GLOBAL ROUTINE INS$REMOVE_VERB =
;  897          1087  1 !+++
;  898          1088  1 !
;  899          1089  1 !   New command interface action routine for REMOVE
;  900          1090  1 !
;  901          1091  1 !---
;  902          1092  2 BEGIN
;  903          1093  2 LOCAL
;  904          1094  2     CMK_ARGLST : VECTOR [1],    ! arguement list for change mode dispatcher
;  905          1095  2     OPEN_STATUS,
;  906          1096  2     STATUS;
;  907          1097  2
;  908          1098  2 !
;  909          1099  2 ! Make delete and remove synonomous actions
;  910          1100  2 !
;  911          1101  2 INS$GL_CTLMSK [INS$V_PROCESS] = CLI$PRESENT( PROCESS_DSC );
;  912          1102  2 IF NOT .INS$GL_CTLMSK [INS$V_PROCESS]
;  913          1103  2 THEN
;  914          1104  3     BEGIN
;  915          1105  3     !
;  916          1106  3     !   Since INSTALL may be installed with CMKRNL privilege,
;  917          1107  3     !   check that user has CMKRNL privilege before allowing
;  918          1108  3     !   any operations other than /PROCESS operations.
;  919          1109  3     !
;  920          1110  3     CMK_ARGLST [0] = 0;
;  921          1111  3     STATUS = $CMKRNL (ROUTIN = INS$CHECK_PRIV, ARGLST = CMK_ARGLST);
;  922          1112  3     IF NOT .STATUS
;  923          1113  3     THEN
;  924          1114  4         BEGIN
;  925          1115  4         SIGNAL (SS$_NOCMKRNL);
;  926          1116  4         RETURN TRUE;
;  927          1117  3         END;
;  928          1118  2     END;
;  929          1119  2
;  930          1120  2 INS$GL_CTLMSK [INS$V_FILSPC] = CLI$PRESENT( FILE_SPEC_DSC );
;  931          1121  2     !
;  932          1122  2     ! Get the file spec from the CLI
;  933          1123  2     !
;  934          1124  2     REPORT( CLI$GET_VALUE ( FILE_SPEC_DSC, INS$GQ_FILDSC) );
;  935          1125  2     OPEN_STATUS = INS$OPEN_FILE ();
;  936          1126  2     IF .OPEN_STATUS
;  937          1127  2     THEN
;  938          1128  3         BEGIN
;  939          1129  3         IF .INS$GL_KFEADR NEQ 0
;  940          1130  3         THEN
;  941          1131  3             STATUS = INS$DELETE ()              ! delete it
;  942          1132  3         ELSE
;  943          1133  3             STATUS = INS$_NOKFEFND;
;  944          1134  3         END
;  945          1135  3
;  946          1136  2     ELSE                                        ! Not opened, try to remove entry
;  947          1137  3         BEGIN
;  948          1138  3         INS$CLOSE_FILE ();
;  949          1139  3         STATUS = INS$DELETE ();         ! INS$DELETE with INS$GL_KFEADR = 0 is a REMOVE
;  950          1140  3         IF .STATUS
;  951          1141  3         THEN
```

```
;  952          1142  4              BEGIN
;  953          1143  4              SIGNAL (.STATUS, 1, INS$GQ_KFERNS);
;  954          1144  4              RETURN TRUE;
;  955          1145  3              END;
;  956          1146  2          END;
;  957          1147  2
;  958          1148  2      IF NOT .STATUS
;  959          1149  2      THEN
;  960          1150  3          BEGIN
;  961          1151  3          IF .STATUS EQL INS$_INTRNLERR
;  962          1152  3          THEN
;  963          1153  3              SIGNAL (((INS$_FAIL AND NOT STS$M_SEVERITY) OR STS$K_WARNING),
;  964          1154  3                  2,REMOVE_DSC,INS$GQ_KFERNS,.STATUS,1,.INS$L_INTRNLERR)
;  965          1155  3          ELSE
;  966          1156  3              SIGNAL (((INS$_FAIL AND NOT STS$M_SEVERITY) OR STS$K_WARNING),
;  967          1157  3                  2,REMOVE_DSC,INS$GQ_KFERNS,.STATUS);
;  968          1158  2          END;
;  969          1159  2
;  970          1160  2 RETURN TRUE;
;  971          1161  1 END;                                    ! routine INS$REMOVE_VERB


                                                  00FC 00000          .ENTRY  INS$REMOVE_VERB, Save R2,R3,R4,R5,R6,R7      ; 1086
                                57 00000000G      00   9E 00002          MOVAB   INS$DELETE, R7                               ;
                                56 00000000G      00   9E 00009          MOVAB   CLI$PRESENT, R6                              ;
                                55      0000'     CF   9E 00010          MOVAB   FILE_SPEC_DSC, R5                            ;
                                54      0000'     CF   9E 00015          MOVAB   INS$GL_CTLMSK, R4                            ;
                                53 00000000G      00   9E 0001A          MOVAB   LIB$SIGNAL, R3                               ;
                                5E                04   C2 00021          SUBL2   #4, SP                                       ;
                                        00A4      C5   9F 00024          PUSHAB  PROCESS_DSC                                  ; 1101
                                66                01   FB 00028          CALLS   #1, CLI$PRESENT                              ;
                64                    01          01   50 F0 0002B          INSV    R0, #1, #1, INS$GL_CTLMSK                    ;
                                      1F          64   01 E0 00030          BBS     #1, INS$GL_CTLMSK, 1$                        ; 1102
                                                  6E   D4 00034          CLRL    CMK_ARGLST                                   ; 1110
                                                  5E   DD 00036          PUSHL   SP                                           ; 1111
                                              0000V    CF   9F 00038          PUSHAB  INS$CHECK_PRIV                               ;
                                   00000000G      00   02 FB 0003C          CALLS   #2, SYS$CMKRNL                               ;
                                                  52   50 D0 00043          MOVL    R0, STATUS                                   ;
                                                  0A   52 E8 00046          BLBS    STATUS, 1$                                   ; 1112
                                                  7E   2804   8F   3C 00049          MOVZWL  #10244, -(SP)                        ; 1115
                                                  63   01 FB 0004E          CALLS   #1, LIB$SIGNAL                               ;
                                                       7A 11 00051          BRB     7$                                           ; 1116
                                                       55 DD 00053 1$:      PUSHL   R5                                           ; 1120
                                                  66   01 FB 00055          CALLS   #1, CLI$PRESENT                              ;
                64                    01          00   50 F0 00058          INSV    R0, #0, #1, INS$GL_CTLMSK                    ;
                                                        1C   A4 9F 0005D          PUSHAB  INS$GQ_FILDSC                        ; 1124
                                                  55   DD 00060          PUSHL   R5                                           ;
                                   00000000G      00   02 FB 00062          CALLS   #2, CLI$GET_VALUE                            ;
                                                  05   50 E8 00069          BLBS    STATUS, 2$                                   ;
                                                  50   DD 0006C          PUSHL   STATUS                                       ;
                                                  63   01 FB 0006E          CALLS   #1, LIB$SIGNAL                               ;
                                        0000V     CF   00 FB 00071 2$:      CALLS   #0, INS$OPEN_FILE                            ; 1125
                                                  16   50 E9 00076          BLBC    OPEN_STATUS, 4$                              ; 1126
                                                        10   A4 D5 00079          TSTL    INS$GL_KFEADR                        ; 1129
```

```
                                      08  13 0007C        BEQL    3$                                   ;
                       67             00  FB 0007E        CALLS   #0, INS$DELETE                       ; 1131
                       52             50  D0 00081        MOVL    R0, STATUS                           ;
                                      23  11 00084        BRB     5$                                   ;
                       52 00000000G   8F  D0 00086 3$:    MOVL    #INS$_NOKFEFND, STATUS               ; 1133
                                      1A  11 0008D        BRB     5$                                   ; 1126
                0000V  CF             00  FB 0008F 4$:    CALLS   #0, INS$CLOSE_FILE                   ; 1138
                       67             00  FB 00094        CALLS   #0, INS$DELETE                       ; 1139
                       52             50  D0 00097        MOVL    R0, STATUS                           ;
                       0F             52  E9 0009A        BLBC    STATUS, 6$                           ; 1140
                                08    A4  9F 0009D        PUSHAB  INS$GQ_KFERNS                        ; 1143
                                      01  DD 000A0        PUSHL   #1                                   ;
                                      52  DD 000A2        PUSHL   STATUS                               ;
                       63             03  FB 000A4        CALLS   #3, LIB$SIGNAL                       ;
                                      39  11 000A7        BRB     9$                                   ; 1144
                       36             52  E8 000A9 5$:    BLBS    STATUS, 9$                           ; 1148
            00000000G  8F             52  D1 000AC 6$:    CMPL    STATUS, #INS$_INTRNLERR              ; 1151
                                      1A  12 000B3        BNEQ    8$                                   ;
                                98    A4  DD 000B5        PUSHL   INS$L_INTRNLERR                      ; 1154
                                      01  DD 000B8        PUSHL   #1                                   ; 1153
                                      52  DD 000BA        PUSHL   STATUS                               ; 1154
                                08    A4  9F 000BC        PUSHAB  INS$GQ_KFERNS                        ; 1153
                                40    A5  9F 000BF        PUSHAB  REMOVE_DSC                           ;
                                      02  DD 000C2        PUSHL   #2                                   ;
                          00000000*   8F  DD 000C4        PUSHL   #<INS$_FAIL&-8>                      ;
                       63             07  FB 000CA        CALLS   #7, LIB$SIGNAL                       ;
                                      13  11 000CD 7$:    BRB     9$                                   ;
                                      52  DD 000CF 8$:    PUSHL   STATUS                               ; 1157
                                08    A4  9F 000D1        PUSHAB  INS$GQ_KFERNS                        ; 1156
                                40    A5  9F 000D4        PUSHAB  REMOVE_DSC                           ;
                                      02  DD 000D7        PUSHL   #2                                   ;
                          00000000*   8F  DD 000D9        PUSHL   #<INS$_FAIL&-8>                      ;
                       63             05  FB 000DF        CALLS   #5, LIB$SIGNAL                       ;
                       50             01  D0 000E2 9$:    MOVL    #1, R0                               ; 1160
                                          04 000E5        RET                                          ; 1161

; Routine Size:  230 bytes,    Routine Base:  $CODE$ + 0757


;  972          1162  1
```

```
;  974          1163  1 %SBTTL  'INS$PURGE_VERB';
;  975          1164  1 GLOBAL ROUTINE INS$PURGE_VERB =
;  976          1165  1 !+++
;  977          1166  1 !
;  978          1167  1 !   New command interface action routine for PURGE
;  979          1168  1 !
;  980          1169  1 !---
;  981          1170  2 BEGIN
;  982          1171  2 LOCAL
;  983          1172  2     CMK_ARGLST : VECTOR [1],    ! arguement list for change mode dispatcher
;  984          1173  2     STATUS;
;  985          1174  2
;  986          1175  2 INS$GL_CTLMSK [INS$V_PROCESS] = CLI$PRESENT( PROCESS_DSC );
;  987          1176  2 IF NOT .INS$GL_CTLMSK [INS$V_PROCESS]
;  988          1177  2 THEN
;  989          1178  3     BEGIN
;  990          1179  3     !
;  991          1180  3     !   Since INSTALL may be installed with CMKRNL privilege,
;  992          1181  3     !   check that user has CMKRNL privilege before allowing
;  993          1182  3     !   any operations other than /PROCESS operations.
;  994          1183  3     !
;  995          1184  3     CMK_ARGLST [0] = 0;
;  996          1185  3     STATUS = $CMKRNL (ROUTIN = INS$CHECK_PRIV, ARGLST = CMK_ARGLST);
;  997          1186  3     IF NOT .STATUS
;  998          1187  3     THEN
;  999          1188  4         BEGIN
; 1000          1189  4         SIGNAL (SS$_NOCMKRNL);
; 1001          1190  4         RETURN TRUE;
; 1002          1191  3         END;
; 1003          1192  2     END;
; 1004          1193  2
; 1005          1194  2 STATUS = INS$PURGE ();
; 1006          1195  2 IF NOT .STATUS
; 1007          1196  2 THEN
; 1008          1197  2     SIGNAL (INS$_FAILED, 1, PURGE_DSC, .STATUS);
; 1009          1198  2
; 1010          1199  2 RETURN TRUE;
; 1011          1200  1 END;                                    ! routine INS$PURGE_VERB


                                                 000C 00000         .ENTRY  INS$PURGE_VERB, Save R2,R3                  ; 1164
                                      53 00000000G   00   9E 00002  MOVAB   LIB$SIGNAL, R3
                                      5E             04   C2 00009  SUBL2   #4, SP
                                               0000'  CF   9F 0000C  PUSHAB  PROCESS_DSC                                 ; 1175
                                 00000000G   00       01   FB 00010  CALLS   #1, CLI$PRESENT
    0000'  CF                       01       01       50   F0 00017  INSV    R0, #1, #1, INS$GL_CTLMSK
                                    1F   0000'  CF       01   E0 0001E  BBS     #1, INS$GL_CTLMSK, 1$                 ; 1176
                                                         6E   D4 00024  CLRL    CMK_ARGLST                            ; 1184
                                                         5E   DD 00026  PUSHL   SP                                    ; 1185
                                                  0000V  CF   9F 00028  PUSHAB  INS$CHECK_PRIV
                                 00000000G   00       02   FB 0002C  CALLS   #2, SYS$CMKRNL
                                             52       50   D0 00033  MOVL    R0, STATUS
                                             0A       52   E8 00036  BLBS    STATUS, 1$                            ; 1186
                                             7E   2804   8F   3C 00039  MOVZWL  #10244, -(SP)                         ; 1189
```

```
                                63              01  FB 0003E          CALLS   #1, LIB$SIGNAL                    ;
                                                1E  11 00041          BRB     2$                               ; 1190
                    00000000G   00              00  FB 00043 1$:      CALLS   #0, INS$PURGE                    ; 1194
                                52              50  D0 0004A          MOVL    R0, STATUS                       ;
                                11              52  E8 0004D          BLBS    STATUS, 2$                       ; 1195
                                                52  DD 00050          PUSHL   STATUS                           ; 1197
                                        0000'   CF  9F 00052          PUSHAB  PURGE_DSC                        ;
                                                01  DD 00056          PUSHL   #1                               ;
                                00000000G       8F  DD 00058          PUSHL   #INS$_FAILED                     ;
                                63              04  FB 0005E          CALLS   #4, LIB$SIGNAL                   ;
                                50              01  D0 00061 2$:      MOVL    #1, R0                           ; 1199
                                                    04 00064          RET                                      ; 1200

; Routine Size:  101 bytes,    Routine Base:  $CODE$ + 083D

; 1012          1201  1
```

```
;1014     1202  1 %SBTTL  'INS$LIST_VERB';
;1015     1203  1 GLOBAL ROUTINE INS$LIST_VERB =
;1016     1204  1 !+++
;1017     1205  1 !
;1018     1206  1 !   New command interface action routine for LIST
;1019     1207  1 !
;1020     1208  1 !---
;1021     1209  2 BEGIN
;1022     1210  2 LOCAL
;1023     1211  2     STATUS;
;1024     1212  2
;1025     1213  2 INS$GL_CTLMSK [INS$V_FILSPC] = CLI$PRESENT( FILE_SPEC_DSC );
;1026     1214  2 INS$GL_CTLMSK [INS$V_FULL] = CLI$PRESENT( %ASCID 'FUL[' );
;1027     1215  2 INS$GL_CTLMSK [INS$V_STRUCTURE] = CLI$PRESENT( %ASCID 'STRUCTURE' );
;1028     1216  2 INS$GL_CTLMSK [INS$V_GLOBAL] = CLI$PRESENT( GLOBAL_DSC );
;1029     1217  2 INS$GL_CTLMSK [INS$V_PROCESS] = CLI$PRESENT( PROCESS_DSC );
;1030     1218  2 INS$GL_CTLMSK [INS$V_PROCESS] = CLI$PRESENT( PROCESS_DSC );
;1031     1219  2
;1032     1220  2 IF NOT .INS$GL_CTLMSK [INS$V_FILSPC]
;1033     1221  2 THEN
;1034     1222  3     BEGIN
;1035     1223  3     !
;1036     1224  3     !   If LIST /GLOBAL then only print global sections
;1037     1225  3     !
;1038     1226  3     IF .INS$GL_CTLMSK [INS$V_GLOBAL]
;1039     1227  3     THEN
;1040     1228  4         BEGIN
;1041     1229  4         STATUS = INS$GLOBAL ();
;1042     1230  4         IF NOT .STATUS
;1043     1231  4         THEN
;1044     1232  4             SIGNAL (((INS$_FAILED AND NOT STS$M_SEVERITY) OR STS$K_WARNING),
;1045     1233  4                                     1, GLOBAL_DSC, .STATUS);
;1046     1234  4         END
;1047     1235  3     ELSE
;1048     1236  4         BEGIN
;1049     1237  4         STATUS = INS$LIST (0);
;1050     1238  4         IF NOT .STATUS
;1051     1239  4         THEN
;1052     1240  4             SIGNAL (((INS$_FAILED AND NOT STS$M_SEVERITY) OR STS$K_WARNING),
;1053     1241  4                                     1, LIST_DSC, .STATUS);
;1054     1242  3         END;
;1055     1243  3     END
;1056     1244  2 ELSE
;1057     1245  3     BEGIN
;1058     1246  3     LOCAL
;1059     1247  3         OPEN_STATUS,
;1060     1248  3         STATUS;
;1061     1249  3
;1062     1250  3     !
;1063     1251  3     ! Get the file spec from the CLI
;1064     1252  3     !
;1065     1253  3     REPORT( CLI$GET_VALUE ( FILE_SPEC_DSC, INS$GQ_FILDSC) );
;1066     1254  3     OPEN_STATUS = INS$OPEN_FILE ();
;1067     1255  4     IF (NOT .OPEN_STATUS)
;1068     1256  3     THEN
;1069     1257  4         BEGIN
;1070     1258  4         INS$CLOSE_FILE ();
```

```
;  1071          1259  4          RETURN TRUE;
;  1072          1260  3          END;
;  1073          1261  3
;  1074          1262  3      IF .INS$GL_KFEADR EQL 0
;  1075          1263  3      THEN
;  1076          1264  4          BEGIN
;  1077          1265  4          SIGNAL (((INS$_FAIL AND NOT STS$M_SEVERITY) OR STS$K_WARNING),
;  1078          1266  4                              2, LIST_DSC,INS$GQ_KFERNS,INS$_NOKFEFND);
;  1079          1267  4          END
;  1080          1268  3      ELSE
;  1081          1269  4          BEGIN
;  1082          1270  4          STATUS = INS$LIST (.INS$GL_KFEADR);
;  1083          1271  4          IF NOT .STATUS
;  1084          1272  4          THEN
;  1085          1273  4              SIGNAL (((INS$_FAILED AND NOT STS$M_SEVERITY) OR STS$K_WARNING),
;  1086          1274  4                                      1, LIST_DSC, .STATUS);
;  1087          1275  4
;  1088          1276  4          IF .INS$GL_CTLMSK [INS$V_GLOBAL]
;  1089          1277  4          THEN
;  1090          1278  5              BEGIN
;  1091          1279  5              STATUS = INS$GLOBAL (.INS$GL_KFEADR);
;  1092          1280  5              IF NOT .STATUS
;  1093          1281  5              THEN
;  1094          1282  7                  SIGNAL (((INS$_FAILED AND NOT STS$M_SEVERITY)
;  1095          1283  5                                      OR STS$K_WARNING), 1, GLOBAL_DSC, .STATUS);
;  1096          1284  4              END;
;  1097          1285  3          END;
;  1098          1286  3
;  1099          1287  3      INS$CLOSE_FILE ();
;  1100          1288  2      END;
;  1101          1289  2
;  1102          1290  2 RETURN TRUE;
;  1103          1291  1 END;                                    ! routine INS$LIST_VERB


                                                                .PSECT  $PLIT$,NOWRT,NOEXE,2

                                                  4C  4C  55  46  001DC P.ACE:   .ASCII  \FULL\                                          ;
                                                    010E0004      001E0 P.ACD:   .LONG   17694724                                        ;
                                                    00000000'     001E4          .ADDRESS P.ACE                                          ;
          00  00  00  45  52  55  54  43  55  52  54  53  001E8 P.ACG:   .ASCII  \STRUCTURE\<0><0><0>                                    ;
                                                    010E0009      001F4 P.ACF:   .LONG   17694729                                        ;
                                                    00000000'     001F8          .ADDRESS P.ACG                                          ;


                                                                .PSECT  $CODE$,NOWRT,2

                                                      01FC 00000          .ENTRY  INS$LIST_VERB, Save R2,R3,R4,R5,R6,R7,R8    ; 1203
                                          58 00000000G  00  9E 00002      MOVAB   INS$LIST, R8                                    ;
                                          57 00000000G  00  9E 00009      MOVAB   INS$GLOBAL, R7                                  ;
                                          56 00000000G  00  9E 00010      MOVAB   LIB$SIGNAL, R6                                  ;
                                          55      0000' CF  9E 00017      MOVAB   GLOBAL_DSC, R5                                  ;
                                          54 00000000G  00  9E 0001C      MOVAB   CLI$PRESENT, R4                                 ;
                                          53      0000' CF  9E 00023      MOVAB   INS$GL_CTLMSK, R3                               ;
                                                    98  A5  9F 00028      PUSHAB  FILE_SPEC_DSC                                   ; 1213
```

```
                                64          01  FB 0002B         CALLS   #1, CLI$PRESENT                     ;
       63           01          00          50  F0 0002E         INSV    R0, #0, #1, INS$GL_CTLMSK           ;
                                      0118  C5  9F 00033         PUSHAB  P.ACD                               ; 1214
                                64          01  FB 00037         CALLS   #1, CLI$PRESENT                     ;
  01   A3           01          02          50  F0 0003A         INSV    R0, #2, #1, INS$GL_CTLMSK+1         ;
                                      012C  C5  9F 00040         PUSHAB  P.ACF                               ; 1215
                                64          01  FB 00044         CALLS   #1, CLI$PRESENT                     ;
  01   A3           01          03          50  F0 00047         INSV    R0, #3, #1, INS$GL_CTLMSK+1         ;
                                            55  DD 0004D         PUSHL   R5                                  ; 1216
                                64          01  FB 0004F         CALLS   #1, CLI$PRESENT                     ;
  01   A3           01          04          50  F0 00052         INSV    R0, #4, #1, INS$GL_CTLMSK+1         ;
                                        3C  A5  9F 00058         PUSHAB  PROCESS_DSC                         ; 1217
                                64          01  FB 0005B         CALLS   #1, CLI$PRESENT                     ;
       63           01          01          50  F0 0005E         INSV    R0, #1, #1, INS$GL_CTLMSK           ;
                                        3C  A5  9F 00063         PUSHAB  PROCESS_DSC                         ; 1218
                                64          01  FB 00066         CALLS   #1, CLI$PRESENT                     ;
       63           01          01          50  F0 00069         INSV    R0, #1, #1, INS$GL_CTLMSK           ;
                                2B          63  E8 0006E         BLBS    INS$GL_CTLMSK, 4$                   ; 1220
                    0C      01  A3          04  E1 00071         BBC     #4, INS$GL_CTLMSK+1, 1$             ; 1226
                                67          00  FB 00076         CALLS   #0, INS$GLOBAL                      ; 1229
                                1E          50  E8 00079         BLBS    STATUS, 3$                          ; 1230
                                            50  DD 0007C         PUSHL   STATUS                              ; 1233
                                            55  DD 0007E         PUSHL   R5                                  ; 1232
                                            0D  11 00080         BRB     2$                                  ;
                                            7E  D4 00082 1$:     CLRL    -(SP)                               ; 1237
                                68          01  FB 00084         CALLS   #1, INS$LIST                        ;
                                10          50  E8 00087         BLBS    STATUS, 3$                          ; 1238
                                            50  DD 0008A         PUSHL   STATUS                              ; 1241
                                        E4  A5  9F 0008C         PUSHAB  LIST_DSC                            ; 1240
                                            01  DD 0008F 2$:     PUSHL   #1                                  ;
                                 00000000*  8F  DD 00091         PUSHL   #<INS$_FAILED&-8>                   ;
                                66          04  FB 00097         CALLS   #4, LIB$SIGNAL                      ;
                                            7C  11 0009A 3$:     BRB     9$                                  ; 1220
                                        1C  A3  9F 0009C 4$:     PUSHAB  INS$GQ_FILDSC                       ; 1253
                                        98  A5  9F 0009F         PUSHAB  FILE_SPEC_DSC                       ;
                     00000000G  00          02  FB 000A2         CALLS   #2, CLI$GET_VALUE                   ;
                                05          50  E8 000A9         BLBS    STATUS, 5$                          ;
                                            50  DD 000AC         PUSHL   STATUS                              ;
                                66          01  FB 000AE         CALLS   #1, LIB$SIGNAL                      ;
                         0000V  CF          00  FB 000B1 5$:     CALLS   #0, INS$OPEN_FILE                   ; 1254
                                5A          50  E9 000B6         BLBC    OPEN_STATUS, 8$                     ; 1255
                                50      10  A3  D0 000B9         MOVL    INS$GL_KFEADR, R0                   ; 1262
                                            19  12 000BD         BNEQ    6$                                  ;
                                 00000000G  8F  DD 000BF         PUSHL   #INS$_NOKFEFND                      ; 1265
                                        08  A3  9F 000C5         PUSHAB  INS$GQ_KFERNS                       ;
                                        E4  A5  9F 000C8         PUSHAB  LIST_DSC                            ;
                                            02  DD 000CB         PUSHL   #2                                  ;
                                 00000000*  8F  DD 000CD         PUSHL   #<INS$_FAIL&-8>                     ;
                                66          05  FB 000D3         CALLS   #5, LIB$SIGNAL                      ;
                                            3B  11 000D6         BRB     8$                                  ; 1262
                                            50  DD 000D8 6$:     PUSHL   R0                                  ; 1270
                                68          01  FB 000DA         CALLS   #1, INS$LIST                        ;
                                52          50  D0 000DD         MOVL    R0, STATUS                          ;
                                10          52  E8 000E0         BLBS    STATUS, 7$                          ; 1271
                                            52  DD 000E3         PUSHL   STATUS                              ; 1274
                                        E4  A5  9F 000E5         PUSHAB  LIST_DSC                            ; 1273
                                            01  DD 000E8         PUSHL   #1                                  ;
```

```
                                00000000*  8F   DD 00CEA          PUSHL   #<INS$_FAILED&-8>                    ;
                                       66  04   FB 000F0          CALLS   #4, LIB$SIGNAL                      ;
                                       1B  01   A3   04   E1 000F3 7$:     BBC     #4, INS$GL_CTLMSK+1, 8$     ; 1276
                                           10   A3   DD 000F8          PUSHL   INS$GL_KFEADR                   ; 1279
                                       67       01   FB 000FB          CALLS   #1, INS$GLOBAL
                                       52       50   D0 000FE          MOVL    R0, STATUS
                                       0F       52   E8 00101          BLBS    STATUS, 8$                      ; 1280
                                                52   DD 00104          PUSHL   STATUS                          ; 1283
                                                55   DD 00106          PUSHL   R5                              ; 1282
                                                01   DD 00108          PUSHL   #1
                                00000000*       8F   DD 0010A          PUSHL   #<INS$_FAILED&-8>
                                       66       04   FB 00110          CALLS   #4, LIB$SIGNAL
                                0000V  CF       00   FB 00113 8$:     CALLS   #0, INS$CLOSE_FILE              ; 1287
                                       50       01   D0 00118 9$:     MOVL    #1, R0                          ; 1290
                                                     04 0011B          RET                                     ; 1291

; Routine Size:  284 bytes,    Routine Base:  $CODE$ + 08A2

; 1104          1292  1
```

```
;  1106          1293  1 %SBTTL  'INS$EXECUTE_IN_EXEC_WITH_R_LOCK';
;  1107          1294  1 GLOBAL ROUTINE INS$EXECUTE_IN_EXEC_WITH_R_LOCK (ROUTINE_NAM, PARAM) =
;  1108          1295  1 !+++
;  1109          1296  1 !
;  1110          1297  1 !   Execute the routine in exec mode
;  1111          1298  1 !
;  1112          1299  1 !---
;  1113          1300  2 BEGIN
;  1114          1301  2 LOCAL
;  1115          1302  2     CME_ARGS : VECTOR [3, LONG],
;  1116          1303  2     STATUS;
;  1117          1304  2
;  1118          1305  2 CME_ARGS [0] = 2;
;  1119          1306  2 CME_ARGS [1] = .ROUTINE_NAM;
;  1120          1307  2 CME_ARGS [2] = .PARAM;
;  1121          1308  2
;  1122          1309  2 STATUS = $CMEXEC (ROUTIN = EXECUTE_WITH_R_LOCK, ARGLST = CME_ARGS);
;  1123          1310  2
;  1124          1311  2 RETURN .STATUS;
;  1125          1312  1 END;              ! Routine INS$EXECUTE_IN_EXEC_WITH_R_LOCK


                                                        .EXTRN  SYS$CMEXEC

                                        0000 00000      .ENTRY  -                                               ; 1294
                                                                INS$EXECUTE_IN_EXEC_WITH_R_LOCK, Save nothi-
                                                                ng
                            5E              08  C2 00002        SUBL2   #8, SP
                                            02  DD 00005        PUSHL   #2                                      ; 1305
                        04  AE          04  AC  7D 00007        MOVQ    ROUTINE_NAM, CME_ARGS+4                 ; 1306
                                            5E  DD 0000C        PUSHL   SP                                      ; 1309
                                    0000V   CF  9F 0000E        PUSHAB  EXECUTE_WITH_R_LOCK
                            00000000G   00      02  FB 00012    CALLS   #2, SYS$CMEXEC
                                                04  00019        RET                                             ; 1312

; Routine Size:  26 bytes,    Routine Base:  $CODE$ + 09BE

; 1126          1313  1
```

```
;  1128          1314  1 %SBTTL  'INS$EXECUTE_IN_KRNL_WITH_W_LOCK';
;  1129          1315  1 GLOBAL ROUTINE INS$EXECUTE_IN_KRNL_WITH_W_LOCK (ROUTINE_NAM, PARAM) =
;  1130          1316  1 !+++
;  1131          1317  1 !
;  1132          1318  1 !   Execute the routine in kernel mode
;  1133          1319  1 !
;  1134          1320  1 !---
;  1135          1321  2 BEGIN
;  1136          1322  2 LOCAL
;  1137          1323  2     CMK_ARGS : VECTOR [3, LONG],
;  1138          1324  2     STATUS;
;  1139          1325  2
;  1140          1326  2 CMK_ARGS [0] = 2;
;  1141          1327  2 CMK_ARGS [1] = .ROUTINE_NAM;
;  1142          1328  2 CMK_ARGS [2] = .PARAM;
;  1143          1329  2
;  1144          1330  2 STATUS = $CMKRNL (ROUTIN = EXECUTE_WITH_W_LOCK, ARGLST = CMK_ARGS);
;  1145          1331  2
;  1146          1332  2 RETURN .STATUS;
;  1147          1333  1 END;              ! Routine INS$EXECUTE_IN_KRNL_WITH_W_LOCK



                                    0000 00000        .ENTRY  -                                               ; 1315
                                                              INS$EXECUTE_IN_KRNL_WITH_W_LOCK, Save nothi-  ;
                                                              ng                                            ;
                          5E          08  C2 00002        SUBL2   #8, SP                                      ;
                                      02  DD 00005        PUSHL   #2                                          ; 1326
                    04    AE      04    AC  7D 00007        MOVQ    ROUTINE_NAM, CMK_ARGS+4                   ; 1327
                                      5E  DD 0000C        PUSHL   SP                                          ; 1330
                                0000V CF  9F 0000E        PUSHAB  EXECUTE_WITH_W_LOCK                         ;
                    00000000G   00        02  FB 00012        CALLS   #2, SYS$CMKRNL                        ;
                                          04  00019        RET                                                 ; 1333

; Routine Size:  26 bytes,    Routine Base:  $CODE$ + 09D8

; 1148          1334  1
```

```
;1150            1335  1 %SBTTL  'execute_with_r_lock';
;1151            1336  1 ROUTINE EXECUTE_WITH_R_LOCK (ROUTINE_NAM, PARAM) =
;1152            1337  1 !+++
;1153            1338  1 !
;1154            1339  1 !   Execute the routine under the protection of a Known file list READ lock.
;1155            1340  1 !
;1156            1341  1 !---
;1157            1342  2 BEGIN
;1158            1343  2 LOCAL
;1159            1344  2     STATUS;
;1160            1345  2
;1161          P 1346  2 STATUS = $ENQW (
;1162          P 1347  2           EFN     = EXE$C_SYSEFN,
;1163          P 1348  2           LKMODE  = LCK$K_PRMODE,
;1164          P 1349  2           LKSB    = INS$G_KF_LKSB,
;1165          P 1350  2           FLAGS   = LCK$M_SYSTEM,
;1166          P 1351  2           RESNAM  = EXE$GQ_KFE_LCKNAM,
;1167          P 1352  2           PARID   = .EXE$GL_SYSID_LOCK,
;1168          P 1353  2           ACMODE  = PSL$C_EXEC
;1169            1354  2                   );
;1170            1355  2
;1171            1356  2 IF .STATUS THEN STATUS = .INS$G_KF_LKSB [LK$W_STATUS];
;1172            1357  2 IF NOT .STATUS THEN RETURN .STATUS;
;1173            1358  2
;1174            1359  2 STATUS = (.ROUTINE_NAM) (.PARAM);
;1175            1360  2
;1176            1361  2 $DEQ (LKID = .INS$G_KF_LKSB [LK$L_LOCKID]);
;1177            1362  2
;1178            1363  2 RETURN .STATUS;
;1179            1364  1 END;                 ! Routine EXECUTE_WITH_R_LOCK


                                                        .EXTRN  SYS$ENQW, SYS$DEQ

                                        0004 00000 EXECUTE_WITH_R_LOCK:
                                                        .WORD   Save R2                                 ;1336
                                7E              01  7D 00002          MOVQ    #1, -(SP)               ;1354
                                                7E  7C 00005          CLRQ    -(SP)                   ;
                                                7E  D4 00007          CLRL    -(SP)                   ;
                                    00000000G   00  DD 00009          PUSHL   EXE$GL_SYSID_LOCK       ;
                                    00000000G   00  9F 0000F          PUSHAB  EXE$GQ_KFE_LCKNAM       ;
                                                10  DD 00015          PUSHL   #16                     ;
                                         0000'  CF  9F 00017          PUSHAB  INS$G_KF_LKSB           ;
                                                03  DD 0001B          PUSHL   #3                      ;
                                    00000000G   8F  DD 0001D          PUSHL   #EXE$C_SYSEFN           ;
                        00000000G   00              0B  FB 00023      CALLS   #11, SYS$ENQW           ;
                                    52              50  D0 0002A      MOVL    R0, STATUS              ;
                                    21              52  E9 0002D      BLBC    STATUS, 1$              ;1356
                                    52         0000'  CF  3C 00030      MOVZWL  INS$G_KF_LKSB, STATUS   ;
                                    19              52  E9 00035      BLBC    STATUS, 1$              ;1357
                                                08  AC  DD 00038      PUSHL   PARAM                   ;1359
                                04  BC              01  FB 0003B      CALLS   #1, @ROUTINE_NAM        ;
                                    52              50  D0 0003F      MOVL    R0, STATUS              ;
                                                7E  7C 00042          CLRQ    -(SP)                   ;1361
                                                7E  D4 00044          CLRL    -(SP)                   ;
                                         0000'  CF  DD 00046          PUSHL   INS$G_KF_LKSB+4         ;
```

```
                 00000000G  00          04  FB 0004A          CALLS   #4, SYS$DEQ                              ;
                            50          52  D0 00051 1$:      MOVL    STATUS, R0                               ; 1363
                                        04  00054            RET                                              ; 1364

; Routine Size:  85 bytes,    Routine Base:  $CODE$ + 09F2

; 1180          1365  1
```

```
;1182        1366  1 %SBTTL  'execute_with_w_lock';
;1183        1367  1 ROUTINE EXECUTE_WITH_W_LOCK (ROUTINE_NAM, PARAM) =
;1184        1368  1 !+++
;1185        1369  1 !
;1186        1370  1 !   Execute the routine under the protection of a Known file list READ lock,
;1187        1371  1 !   and an Install WRITE lock.  During the execution of the routine, the
;1188        1372  1 !   known file list lock must be converted to exclusive write.
;1189        1373  1 !
;1190        1374  1 !---
;1191        1375  2 BEGIN
;1192        1376  2 LOCAL
;1193        1377  2     STATUS;
;1194        1378  2
;1195        1379  2 !
;1196        1380  2 !   Take out INSTALL lock.  This Entitles holder to convert the KFE Read lock
;1197        1381  2 !   to a Write lock later.
;1198        1382  2 !
;1199      P 1383  2 STATUS = $ENQW (
;1200      P 1384  2          EFN     = EXE$C_SYSEFN,
;1201      P 1385  2          LKMODE  = LCK$K_PRMODE,
;1202      P 1386  2          LKSB    = INS$G_INS_LKSB,
;1203      P 1387  2          FLAGS   = LCK$M_SYSTEM,
;1204      P 1388  2          RESNAM  = INS$Q_LCKNAM,
;1205      P 1389  2          PARID   = .EXE$GL_SYSID_LOCK,
;1206      P 1390  2          ACMODE  = PSL$C_EXEC
;1207        1391  2                    );
;1208        1392  2
;1209        1393  2 IF .STATUS THEN STATUS = .INS$G_INS_LKSB [LK$W_STATUS];
;1210        1394  2 IF NOT .STATUS THEN RETURN .STATUS;
;1211        1395  2
;1212        1396  2 !
;1213        1397  2 !   Take out KFE Read lock under protection of INSTALL lock.
;1214        1398  2 !   This permits option to later convert lock to a Write lock.
;1215        1399  2 !
;1216      P 1400  2 STATUS = $ENQW (
;1217      P 1401  2          EFN     = EXE$C_SYSEFN,
;1218      P 1402  2          LKMODE  = LCK$K_PRMODE,
;1219      P 1403  2          LKSB    = INS$G_KF_LKSB,
;1220      P 1404  2          FLAGS   = LCK$M_SYSTEM,
;1221      P 1405  2          RESNAM  = EXE$GQ_KFE_LCKNAM,
;1222      P 1406  2          PARID   = .EXE$GL_SYSID_LOCK,
;1223      P 1407  2          ACMODE  = PSL$C_EXEC
;1224        1408  2                    );
;1225        1409  2
;1226        1410  2 IF .STATUS THEN STATUS = .INS$G_KF_LKSB [LK$W_STATUS];
;1227        1411  2 IF NOT .STATUS THEN RETURN .STATUS;
;1228        1412  2
;1229        1413  2 STATUS = (.ROUTINE_NAM) (.PARAM);
;1230        1414  2
;1231        1415  2 $DEQ (LKID = .INS$G_KF_LKSB [LK$L_LOCKID]);
;1232        1416  2 $DEQ (LKID = .INS$G_INS_LKSB [LK$L_LOCKID]);
;1233        1417  2
;1234        1418  2 RETURN .STATUS;
;1235        1419  1 END;                 ! Routine EXECUTE_WITH_W_LOCK
```

```
                                00FC 00000 EXECUTE_WITH_W_LOCK:
                                           .WORD   Save R2,R3,R4,R5,R6,R7                  ; 1367
57 00000000G   00   9E 00002              MOVAB   EXE$GL_SYSID_LOCK, R7
56 00000000G   00   9E 00009              MOVAB   SYS$DEQ, R6
55 00000000G   00   9E 00010              MOVAB   SYS$ENQW, R5
54 00000000G   8F   D0 00017              MOVL    #EXE$C_SYSEFN, R4
53     0000'   CF   9E 0001E              MOVAB   INS$G_INS_LKSB, R3
7E             01   7D 00023              MOVQ    #1, -(SP)                               ; 1391
               7E   7C 00026              CLRQ    -(SP)
               7E   D4 00028              CLRL    -(SP)
               67   DD 0002A              PUSHL   EXE$GL_SYSID_LOCK
         0000'   CF   9F 0002C              PUSHAB  INS$Q_LCKNAM
               10   DD 00030              PUSHL   #16
               53   DD 00032              PUSHL   R3
               03   DD 00034              PUSHL   #3
               54   DD 00036              PUSHL   R4
65             0B   FB 00038              CALLS   #11, SYS$ENQW
52             50   D0 0003B              MOVL    R0, STATUS
4C             52   E9 0003E              BLBC    STATUS, 1$                              ; 1393
52             63   3C 00041              MOVZWL  INS$G_INS_LKSB, STATUS
46             52   E9 00044              BLBC    STATUS, 1$                              ; 1394
7E             01   7D 00047              MOVQ    #1, -(SP)                               ; 1408
               7E   7C 0004A              CLRQ    -(SP)
               7E   D4 0004C              CLRL    -(SP)
               67   DD 0004E              PUSHL   EXE$GL_SYSID_LOCK
    00000000G   00   9F 00050              PUSHAB  EXE$GQ_KFE_LCKNAM
               10   DD 00056              PUSHL   #16
         F8    A3   9F 00058              PUSHAB  INS$G_KF_LKSB
               03   DD 0005B              PUSHL   #3
               54   DD 0005D              PUSHL   R4
            65      0B   FB 0005F              CALLS   #11, SYS$ENQW
            52      50   D0 00062              MOVL    R0, STATUS
            25      52   E9 00065              BLBC    STATUS, 1$                          ; 1410
            52   F8 A3   3C 00068              MOVZWL  INS$G_KF_LKSB, STATUS
            1E      52   E9 0006C              BLBC    STATUS, T$                          ; 1411
               08   AC   DD 0006F              PUSHL   PARAM                               ; 1413
      04    BC           01   FB 00072              CALLS   #1, @ROUTINE_NAM
            52           50   D0 00076              MOVL    R0, STATUS
                         7E   7C 00079              CLRQ    -(SP)                           ; 1415
                         7E   D4 0007B              CLRL    -(SP)
                      FC A3   DD 0007D              PUSHL   INS$G_KF_LKSB+4
            66           04   FB 00080              CALLS   #4, SYS$DEQ
                         7E   7C 00083              CLRQ    -(SP)                           ; 1416
                         7E   D4 00085              CLRL    -(SP)
                      04 A3   DD 00087              PUSHL   INS$G_INS_LKSB+4
            66           04   FB 0008A              CALLS   #4, SYS$DEQ
            50           52   D0 0008D 1$:          MOVL    STATUS, R0                      ; 1418
                              04 00090              RET                                     ; 1419

; Routine Size:  145 bytes,    Routine Base:  $CODE$ + 0A47

; 1236          1420  1
```

```
;  1238          1421  1 %SBTTL  'INS$CNVRT_KF_LOCK';
;  1239          1422  1 GLOBAL ROUTINE INS$CNVRT_KF_LOCK (MODE) =
;  1240          1423  1 !+++
;  1241          1424  1 !
;  1242          1425  1 !   Convert the currently held KF lock to a different mode.
;  1243          1426  1 !
;  1244          1427  1 !---
;  1245          1428  2 BEGIN
;  1246          1429  2 LOCAL
;  1247          1430  2     STATUS;
;  1248          1431  2
;  1249          1432  2 !
;  1250          1433  2 !
;  1251          1434  2 !
;  1252        P 1435  2 STATUS = $ENQW (
;  1253        P 1436  2          EFN     = EXE$C_SYSEFN,
;  1254        P 1437  2          LKMODE  = .MODE,
;  1255        P 1438  2          LKSB    = INS$G_KF_LKSB,
;  1256        P 1439  2          FLAGS   = LCK$M_CONVERT,
;  1257        P 1440  2          RESNAM  = EXE$GQ_KFE_LCKNAM,
;  1258        P 1441  2          PARID   = .EXE$GL_SYSID_LOCK,
;  1259        P 1442  2          ACMODE  = PSL$C_EXEC
;  1260          1443  2                    );
;  1261          1444  2
;  1262          1445  2 IF .STATUS THEN STATUS = .INS$G_KF_LKSB [LK$W_STATUS];
;  1263          1446  2
;  1264          1447  2 RETURN .STATUS;
;  1265          1448  2
;  1266          1449  1 END;              ! Routine INS$CNVRT_KF_LOCK



                                            0000 00000          .ENTRY  INS$CNVRT_KF_LOCK, Save nothing                 ; 1422
                                  7E        01  7D 00002          MOVQ    #1, -(SP)                                       ; 1443
                                            7E  7C 00005          CLRQ    -(SP)                                           ;
                                            7E  D4 00007          CLRL    -(SP)                                           ;
                                00000000G   00  DD 00009          PUSHL   EXE$GL_SYSID_LOCK                               ;
                                00000000G   00  9F 0000F          PUSHAB  EXE$GQ_KFE_LCKNAM                               ;
                                            02  DD 00015          PUSHL   #2                                              ;
                                    0000'   CF  9F 00017          PUSHAB  INS$G_KF_LKSB                                   ;
                                      04    AC  DD 0001B          PUSHL   MODE                                            ;
                                00000000G   8F  DD 0001E          PUSHL   #EXE$C_SYSEFN                                   ;
                    00000000G   00          0B  FB 00024          CALLS   #11, SYS$ENQW                                   ;
                                05          50  E9 0002B          BLBC    STATUS, 1$                                      ; 1445
                                50      0000'   CF  3C 0002E          MOVZWL  INS$G_KF_LKSB, STATUS                           ;
                                                04 00033 1$:      RET                                                     ; 1449

; Routine Size:  52 bytes,    Routine Base:  $CODE$ + 0AD8

;  1267          1450  1
```

```
;1269     1451  1 %SBTTL  'INS$CVT_DIR';
;1270     1452  1 GLOBAL ROUTINE INS$CVT_DIR (DESC) : NOVALUE =
;1271     1453  2 BEGIN
;1272     1454  2 !++
;1273     1455  2 ! FUNCTIONAL DESCRIPTION:
;1274     1456  2 !
;1275     1457  2 !       Convert square brackets in file spec string to angle brackets. Also, eliminate
;1276     1458  2 !       middle brackets if they exist because the directory was rooted.
;1277     1459  2 !--
;1278     1460  2 MAP
;1279     1461  2     DESC : REF $BBLOCK;
;1280     1462  2
;1281     1463  2 LOCAL
;1282     1464  2     PTR,
;1283     1465  2     PTR2;
;1284     1466  2 !
;1285     1467  2 ! Convert "[" and "]" to "<" and ">"
;1286     1468  2 !
;1287     1469  2 PTR = CH$FIND_CH(.DESC[DSC$W_LENGTH],.DESC[DSC$A_POINTER],'[');
;1288     1470  2 IF NOT CH$FAIL(.PTR)
;1289     1471  2 THEN
;1290     1472  3     BEGIN
;1291     1473  3     CH$WCHAR('<',.PTR);
;1292     1474  3     PTR = CH$FIND_CH(.DESC[DSC$W_LENGTH]-(.PTR-.DESC[DSC$A_POINTER]),.PTR,'[');
;1293     1475  3     IF NOT CH$FAIL(.PTR)
;1294     1476  3     THEN
;1295     1477  3         CH$WCHAR('<',.PTR);
;1296     1478  2     END;
;1297     1479  2
;1298     1480  2 PTR = CH$FIND_CH(.DESC[DSC$W_LENGTH],.DESC[DSC$A_POINTER],']');
;1299     1481  2 IF NOT CH$FAIL(.PTR)
;1300     1482  2 THEN
;1301     1483  3     BEGIN
;1302     1484  3     CH$WCHAR('>',.PTR);
;1303     1485  3     PTR = CH$FIND_CH(.DESC[DSC$W_LENGTH]-(.PTR-.DESC[DSC$A_POINTER]),.PTR,']');
;1304     1486  3     IF NOT CH$FAIL(.PTR)
;1305     1487  3     THEN
;1306     1488  3         CH$WCHAR('>',.PTR);
;1307     1489  2     END;
;1308     1490  2
;1309     1491  2 !
;1310     1492  2 ! If the device was rooted, there should be a middle set of brackets in
;1311     1493  2 ! the string. Compress them out of the string.
;1312     1494  2 !
;1313     1495  2
;1314     1496  2 PTR = CH$FIND_CH(.DESC[DSC$W_LENGTH],.DESC[DSC$A_POINTER],'>');
;1315     1497  2 IF NOT CH$FAIL(.PTR)
;1316     1498  2 THEN
;1317     1499  3     BEGIN
;1318     1500  3     PTR2 = CH$FIND_CH(.DESC[DSC$W_LENGTH]-(.PTR+1-.DESC[DSC$A_POINTER]),.PTR+1,'<');
;1319     1501  3     IF NOT CH$FAIL(.PTR2)
;1320     1502  3     THEN
;1321     1503  4         BEGIN
;1322     1504  4         CH$MOVE(.DESC[DSC$W_LENGTH]-(.PTR2+1-.DESC[DSC$A_POINTER]),.PTR2+1,.PTR);
;1323     1505  4         DESC[DSC$W_LENGTH] = .DESC[DSC$W_LENGTH]-(.PTR2-.PTR+1);
;1324     1506  3         END;
;1325     1507  2     END;
```

```
;  1326          1508  2
;  1327          1509  2 RETURN;
;  1328          1510  1 END;


                                                01FC 00000          .ENTRY  INS$CVT_DIR, Save R2,R3,R4,R5,R6,R7,R8      ; 1452
                                    58      04  AC   D0 00002          MOVL    DESC, R8                                    ; 1469
                                    52          68   3C 00006          MOVZWL  (R8), R2                                    ;
                                    53      04  A8   D0 00009          MOVL    4(R8), R3                                   ;
                              63    52      5B  8F   3A 0000D          LOCC    #91, R2, (R3)                               ;
                                                02   12 00012          BNEQ    1$                                          ;
                                                51   D4 00014          CLRL    R1                                          ;
                                    56          51   D0 00016 1$:      MOVL    R1, PTR                                     ;
                                                1B   13 00019          BEQL    3$                                          ; 1470
                                    66          3C   90 0001B          MOVB    #60, (PTR)                                  ; 1473
                              50    53          56   C3 0001E          SUBL3   PTR, R3, R0                                 ; 1474
                                    50          52   C0 00022          ADDL2   R2, R0                                      ;
                              66    50      5B  8F   3A 00025          LOCC    #91, R0, (PTR)                              ;
                                                02   12 0002A          BNEQ    2$                                          ;
                                                51   D4 0002C          CLRL    R1                                          ;
                                    56          51   D0 0002E 2$:      MOVL    R1, PTR                                     ;
                                                03   13 00031          BEQL    3$                                          ; 1475
                                    66          3C   90 00033          MOVB    #60, (PTR)                                  ; 1477
                              63    52      5D  8F   3A 00036 3$:      LOCC    #93, R2, (R3)                               ; 1480
                                                02   12 0003B          BNEQ    4$                                          ;
                                                51   D4 0003D          CLRL    R1                                          ;
                                    56          51   D0 0003F 4$:      MOVL    R1, PTR                                     ;
                                                1B   13 00042          BEQL    6$                                          ; 1481
                                    66          3E   90 00044          MOVB    #62, (PTR)                                  ; 1484
                              50    53          56   C3 00047          SUBL3   PTR, R3, R0                                 ; 1485
                                    50          52   C0 0004B          ADDL2   R2, R0                                      ;
                              66    50      5D  8F   3A 0004E          LOCC    #93, R0, (PTR)                              ;
                                                02   12 00053          BNEQ    5$                                          ;
                                                51   D4 00055          CLRL    R1                                          ;
                                    56          51   D0 00057 5$:      MOVL    R1, PTR                                     ;
                                                03   13 0005A          BEQL    6$                                          ; 1486
                                    66          3E   90 0005C          MOVB    #62, (PTR)                                  ; 1488
                              63    52          3E   3A 0005F 6$:      LOCC    #62, R2, (R3)                               ; 1496
                                                02   12 00063          BNEQ    7$                                          ;
                                                51   D4 00065          CLRL    R1                                          ;
                                    56          51   D0 00067 7$:      MOVL    R1, PTR                                     ;
                                                39   13 0006A          BEQL    9$                                          ; 1497
                              50    53          56   C3 0006C          SUBL3   PTR, R3, R0                                 ; 1500
                                    50      FF  A042 9E 00070          MOVAB   -1(R0)[R2], R0                              ;
                                    51          56   D0 00075          MOVL    PTR, R1                                     ;
                        01    A1    50          3C   3A 00078          LOCC    #60, R0, 1(R1)                              ;
                                                02   12 0007D          BNEQ    8$                                          ;
                                                51   D4 0007F          CLRL    R1                                          ;
                                    57          51   D0 00081 8$:      MOVL    R1, PTR2                                    ;
                                                1F   13 00084          BEQL    9$                                          ; 1501
                              50    53          57   C3 00086          SUBL3   PTR2, R3, R0                                ; 1504
                                    50      FF  A042 9E 0008A          MOVAB   -1(R0)[R2], R0                              ;
                                    51          57   D0 0008F          MOVL    PTR2, R1                                    ;
                              66    01    A1    50   28 00092          MOVC3   R0, 1(R1), (PTR)                            ;
```

```
                 51          57          56  C3 00097          SUBL3   PTR, PTR2, R1                   ; 1505
                             50          68  3C 0009B          MOVZWL  (R8), R0                        ;
                             50          51  C2 0009E          SUBL2   R1, R0                          ;
                 68          50          01  A3 000A1          SUBW3   #1, R0, (R8)                    ;
                                             04 000A5 9$:      RET                                     ; 1510

; Routine Size:  166 bytes,    Routine Base:  $CODE$ + 0B0C
```

```
;  1330          1511  1 %SBTTL  'INS$OPEN_FILE';
;  1331          1512  1 GLOBAL ROUTINE INS$OPEN_FILE =
;  1332          1513  2 BEGIN
;  1333          1514  2
;  1334          1515  2 OWN
;  1335          1516  2     INS$L_KFEESS : LONG,                  ! buffer for expanded name string
;  1336          1517  2     INS$L_KFERSS : LONG;                  ! buffer for resultant name string
;  1337          1518  2
;  1338          1519  2 LOCAL
;  1339          1520  2     VERSION_LENGTH,                       ! holds length of version string in RNS
;  1340          1521  2     DEVNAM : BBLOCK [65],                 ! scratch device name buffer
;  1341          1522  2     DEV_DSC : $BBLOCK [DSC$C_S_BLN],      ! device name descriptor
;  1342          1523  2     ITMLST : VECTOR [4,LONG]
;  1343          1524  2         PRESET ( [0] = DVI$_LOGVOLNAM ^ 16 + 64,
;  1344          1525  2                  [3] = 0),
;  1345          1526  2     DESC : $BBLOCK [DSC$C_S_BLN],         ! buffer for result name string desc.
;  1346          1527  2     OPEN_STATUS,
;  1347          1528  2     PTR,
;  1348          1529  2     STATUS;
;  1349          1530  2
;  1350          1531  2 !
;  1351          1532  2 ! Allocate RSS and ESS if not done so yet.
;  1352          1533  2 !
;  1353          1534  2 IF .INS$L_KFEESS EQL 0
;  1354          1535  3 THEN BEGIN
;  1355          1536  3     STATUS = LIB$GET_VM(%REF(NAM$C_MAXRSS),INS$L_KFEESS);
;  1356          1537  3     IF NOT .STATUS
;  1357          1538  3         THEN (SIGNAL(.STATUS); RETURN .STATUS);
;  1358          1539  2     END;
;  1359          1540  2 IF .INS$L_KFERSS EQL 0
;  1360          1541  3 THEN BEGIN
;  1361          1542  3     STATUS = LIB$GET_VM(%REF(NAM$C_MAXRSS),INS$L_KFERSS);
;  1362          1543  3     IF NOT .STATUS
;  1363          1544  3         THEN (SIGNAL(.STATUS); RETURN .STATUS);
;  1364          1545  2     END;
;  1365          1546  2 !+++
;  1366          1547  2 !
;  1367          1548  2 !   Prepare to OPEN the file
;  1368          1549  2 !
;  1369          1550  2 !---
;  1370          1551  2
;  1371        P 1552  2 $FAB_INIT (
;  1372        P 1553  2           FAB = INS$G_KFEFAB,
;  1373        P 1554  2           NAM = INS$G_KFENAM,
;  1374        P 1555  2           SHR = <GET,PUT,UPI>,
;  1375        P 1556  2           DNM = 'SYS$SYSTEM:.EXE',
;  1376        P 1557  2           FNA = .INS$GQ_FILDSC [DSC$A_POINTER],
;  1377        P 1558  2           FNS = .INS$GQ_FILDSC [DSC$W_LENGTH],
;  1378        P 1559  2           FOP = <UFO, KFO>,
;  1379        P 1560  2           RTV = -1
;  1380          1561  2           );
;  1381          1562  2 INS$G_KFEFAB [FAB$L_CTX] = 0;    ! Zero CTX longword in the FAB to
;  1382          1563  2                                  ! determine if KFE address is returned by OPEN
;  1383          1564  2
;  1384        P 1565  2 $NAM_INIT (
;  1385        P 1566  2           NAM = INS$G_KFENAM,
;  1386        P 1567  2           ESA = .INS$L_KFEESS,
```

```
;  1387        P 1568  2          ESS = NAM$C_MAXRSS,
;  1388        P 1569  2          RSA = .INS$C_KFERSS,
;  1389        P 1570  2          RSS = NAM$C_MAXRSS,
;  1390        P 1571  2          NOP = <NOCONCEAL>
;  1391          1572  2          );
;  1392          1573  2
;  1393          1574  2 IF .INS$GL_CTLMSK [INS$V_WRITABLE]
;  1394          1575  2     THEN INS$G_KFEFAB [FAB$B_FAC] = FAB$M_PUT;
;  1395          1576  2
;  1396          1577  2 !+++
;  1397          1578  2 !
;  1398          1579  2 !   Open the file
;  1399          1580  2 !
;  1400          1581  2 !---
;  1401          1582  2
;  1402          1583  2 OPEN_STATUS = $OPEN (FAB = INS$G_KFEFAB);
;  1403          1584  2 !
;  1404          1585  2 !   Disallow the installation of sequentially loaded files
;  1405          1586  2 !   such as network and magtape.
;  1406          1587  2 !
;  1407          1588  2 IF .OPEN_STATUS
;  1408          1589  2 THEN
;  1409          1590  3     BEGIN
;  1410          1591  3     BIND
;  1411          1592  3         DEV = INS$G_KFEFAB [FAB$L_DEV] : $BBLOCK;
;  1412          1593  3
;  1413          1594  3     IF .DEV [DEV$V_NET] OR .DEV [DEV$V_SQD]
;  1414          1595  3     THEN
;  1415          1596  3         OPEN_STATUS = INS$_NOLOAD;
;  1416          1597  2     END;
;  1417          1598  2
;  1418          1599  2 IF NOT .OPEN_STATUS      ! open error
;  1419          1600  2 THEN
;  1420          1601  3     BEGIN
;  1421          1602  3     INS$G_KFEFAB[FAB$L_CTX] = INS$_OPENIN;
;  1422          1603  3     IF .COMMAND_MODE EQL OLD
;  1423          1604  3     THEN
;  1424          1605  3         IF (STR$COMPARE (HEL_DSC, INS$GQ_FILDSC) EQL 0) OR
;  1425          1606  4            (STR$COMPARE (HELP_DSC, INS$GQ_FILDSC) EQL 0)
;  1426          1607  3         THEN
;  1427          1608  4             BEGIN
;  1428          1609  4             SIGNAL (INS$_HELP);
;  1429          1610  3             END;
;  1430          1611  3     END
;  1431          1612  2 ELSE
;  1432          1613  2     INS$GL_KFECHAN = .INS$G_KFEFAB [FAB$L_STV]; ! save the channel
;  1433          1614  2
;  1434          1615  2 !+++
;  1435          1616  2 !
;  1436          1617  2 !   Save the resultant name string. Extract the device name and convert
;  1437          1618  2 !   it into the volume logical name.
;  1438          1619  2 !
;  1439          1620  2 !---
;  1440          1621  2
;  1441          1622  2 CH$MOVE(DSC$C_S_BLN,UTIL$GETFILENAME(INS$G_KFEFAB),DESC);
;  1442          1623  2 IF .OPEN_STATUS
;  1443          1624  2 THEN
```

```
;  1444          1625  3     VERSION_LENGTH = .DESC[DSC$W_LENGTH] - (
;  1445          1626  3         .INS$G_KFENAM [NAM$B_DEV] +
;  1446          1627  3         .INS$G_KFENAM [NAM$B_DIR] +
;  1447          1628  3         .INS$G_KFENAM [NAM$B_NAME] +
;  1448          1629  3         .INS$G_KFENAM [NAM$B_TYPE])
;  1449          1630  2 ELSE
;  1450          1631  2     VERSION_LENGTH = 0;
;  1451          1632  2 !
;  1452          1633  2 ! Extract device name from RNS
;  1453          1634  2 !
;  1454          1635  2 PTR = CH$FIND_CH(.DESC[DSC$W_LENGTH],.DESC[DSC$A_POINTER],':');
;  1455          1636  2 IF NOT CH$FAIL(.PTR)
;  1456          1637  2 THEN
;  1457          1638  3     BEGIN
;  1458          1639  3     DEV_DSC[DSC$A_POINTER] = DEVNAM;                    ! Setup devnam descriptor
;  1459          1640  3     CH$WCHAR(%C':',.DEV_DSC[DSC$A_POINTER]);            ! Indicate device to GETDVI
;  1460          1641  3     CH$MOVE(.PTR-.DESC[DSC$A_POINTER],.DESC[DSC$A_POINTER],
;  1461          1642  3                 .DEV_DSC[DSC$A_POINTER] + 1);       ! Move in device name
;  1462          1643  3     DEV_DSC[DSC$W_LENGTH] = .PTR - .DESC[DSC$A_POINTER] + 1; ! Count underscore
;  1463          1644  3
;  1464          1645  3     ITMLST [1] = .INS$GQ_KFERNS[DSC$A_POINTER];         ! Volume name buffer
;  1465          1646  3     ITMLST [2] = INS$GQ_KFERNS[DSC$W_LENGTH];           ! Place for string length
;  1466          1647  3     SYS$GETDVIW(0,0,DEV_DSC,ITMLST,0,0,0,0);            ! Get volume name string
;  1467          1648  3
;  1468          1649  3     !
;  1469          1650  3     ! Move remainder of RNS (except version string) behind volume name
;  1470          1651  3     !
;  1471          1652  3     PTR = CH$MOVE(.DESC[DSC$W_LENGTH]-(.PTR-.DESC[DSC$A_POINTER]),
;  1472          1653  3                 .PTR,.INS$GQ_KFERNS[DSC$A_POINTER]+
;  1473          1654  3                 .INS$GQ_KFERNS[DSC$W_LENGTH]);
;  1474          1655  3     INS$GQ_KFERNS[DSC$W_LENGTH] = .PTR - .INS$GQ_KFERNS[DSC$A_POINTER] -
;  1475          1656  3                 .VERSION_LENGTH;
;  1476          1657  3     END
;  1477          1658  3 !
;  1478          1659  3 ! Use RNS string less version length
;  1479          1660  3 !
;  1480          1661  2 ELSE
;  1481          1662  3     BEGIN
;  1482          1663  3     CH$MOVE(.DESC[DSC$W_LENGTH],.DESC[DSC$A_POINTER],
;  1483          1664  3                 .INS$GQ_KFERNS[DSC$A_POINTER]);
;  1484          1665  3     INS$GQ_KFERNS [DSC$W_LENGTH] = .DESC[DSC$W_LENGTH] - .VERSION_LENGTH;
;  1485          1666  2     END;
;  1486          1667  2
;  1487          1668  2 !
;  1488          1669  2 ! Convert brackets
;  1489          1670  2 !
;  1490          1671  2 INS$CVT_DIR(INS$GQ_KFERNS);
;  1491          1672  2
;  1492          1673  2 !
;  1493          1674  2 ! If open had not succeeded, report the error and return open status.
;  1494          1675  2 !
;  1495          1676  2 IF NOT .OPEN_STATUS
;  1496          1677  2 THEN
;  1497          1678  3     BEGIN
;  1498          1679  3     IF .INS$G_KFEFAB [FAB$L_STS] NEQ RMS$_EOF
;  1499          1680  3     THEN
;  1500          1681  3         SIGNAL (.INS$G_KFEFAB [FAB$L_CTX], 1, INS$GQ_KFERNS,
```

```
; 1501        1682 3                 .INS$G_KFEFAB [FAB$L_STS], .INS$G_KFEFAB [FAB$L_STV]);
; 1502        1683 3
; 1503        1684 3      INS$GL_KFEADR = 0;                  ! No KFE to point to in this case
; 1504        1685 3      RETURN .OPEN_STATUS;
; 1505        1686 2      END;
; 1506        1687 2 !
; 1507        1688 2 !        Store the KFE address returned by RMS OPEN.
; 1508        1689 2 !        After a replace or create operation, those routines will deposit the
; 1509        1690 2 !        new KFE address in ins$gl_kfeadr in case /LIST was requested
; 1510        1691 2 !
; 1511        1692 2 INS$GL_KFEADR = .INS$G_KFEFAB [FAB$L_CTX];
; 1512        1693 2
; 1513        1694 2 RETURN TRUE;
; 1514        1695 1 END;            ! routine INS$OPEN_FILE


                                                        .PSECT  $PLIT$,NOWRT,NOEXE,2

                              002C0040  001FC P.ACH:    .LONG   2883648                         ;
                                   00#  00200           .BYTE   0[8]                            ;
                              00000000  00208           .LONG   0                               ;
45 58 45 2E 3A 4D 45 54 53 59 53 24 53 59 53  0020C P.ACI:  .ASCII  \SYS$SYSTEM:.EXE\           ;

                                                        .PSECT  $OWN$,NOEXE,2

                                        00034 INS$L_KFEESS:
                                                        .BLKB   4
                                        00038 INS$L_KFERSS:
                                                        .BLKB   4

                                              $RMS_PTR=         INS$G_KFEFAB
                                              $RMS_PTR=         INS$G_KFENAM
                                              DEV=              INS$G_KFEFAB+64
                                                        .EXTRN  SYS$OPEN

                                                        .PSECT  $CODE$,NOWRT,2

                                   0FFC 00000           .ENTRY  INS$OPEN_FILE, Save R2,R3,R4,R5,R6,R7,R8,-   ; 1512
                                                                R9,R10,R11                           ;
                5B       0000'  CF  9E 00002            MOVAB   INS$L_KFEESS, R11                    ;
                5A       0000'  CF  9E 00007            MOVAB   INS$GQ_KFERNS, R10                   ;
                5E         98   AE  9E 0000C            MOVAB   -104(SP), SP                         ;
   0C   AE   0000'  CF          10  28 00010            MOVC3   #16, P.ACH, ITMLST                   ; 1525
                           6B   D5 00017                TSTL    INS$L_KFEESS                         ; 1534
                           17   12 00019                BNEQ    1$                                   ;
                           5B   DD 0001B                PUSHL   R11                                  ; 1536
           04   AE         FF   8F  9A 0001D            MOVZBL  #255, 4(SP)                          ;
                           04   AE  9F 00022            PUSHAB  4(SP)                                ;
      00000000G  00             02  FB 00025            CALLS   #2, LIB$GET_VM                       ;
                52              50  D0 0002C            MOVL    R0, STATUS                           ;
                1D              52  E9 0002F            BLBC    STATUS, 2$                           ; 1537
                           04   AB  D5 00032 1$:        TSTL    INS$L_KFERSS                         ; 1540
                                25  12 00035            BNEQ    3$                                   ;
                           04   AB  9F 00037            PUSHAB  INS$L_KFERSS                         ; 1542
           04   AE         FF   8F  9A 0003A            MOVZBL  #255, 4(SP)                          ;
                           04   AE  9F 0003F            PUSHAB  4(SP)                                ;
```

```
                00000000G  00          02  FB 00042          CALLS   #2, LIB$GET_VM                                  ;
                           52          50  D0 00049          MOVL    R0, STATUS                                      ;
                           0D          52  E8 0004C          BLBS    STATUS, 3$                                      ; 1543
                                       52  DD 0004F 2$:      PUSHL   STATUS                                          ; 1544
                00000000G  00          01  FB 00051          CALLS   #1, LIB$SIGNAL                                  ;
                           50          52  D0 00058          MOVL    STATUS, R0                                      ;
                                       04  0005B             RET                                                     ;
    0050  8F       00      6E          00  2C 0005C 3$:      MOVC5   #0, (SP), #0, #80, $RMS_PTR                     ; 1561
                                00B0   CA      00063                                                                 ;
                        00B0   CA  5003    8F  B0 00066          MOVW    #20483, $RMS_PTR                            ;
                        00B4   CA  40020000 8F D0 0006D          MOVL    #1073872896, $RMS_PTR+4                     ;
                        00C6   CA  4302    8F  B0 00076          MOVW    #17154, $RMS_PTR+22                         ;
                        00CC   CA          01  8E 0007D          MNEGB   #1, $RMS_PTR+28                             ;
                        00CF   CA          02  90 00082          MOVB    #2, $RMS_PTR+31                             ;
                        00D8   CA    94    AA  9E 00087          MOVAB   INS$G_KFENAM, $RMS_PTR+40                   ;
                        00DC   CA    18    AA  D0 0008D          MOVL    INS$GQ_FILDSC+4, $RMS_PTR+44                ;
                        00E0   CA  0000'   CF  9E 00093          MOVAB   P.ACI, $RMS_PTR+48                          ;
                        00E4   CA    14    AA  90 0009A          MOVB    INS$GQ_FILDSC, $RMS_PTR+52                  ;
                        00E5   CA          0F  90 000A0          MOVB    #15, $RMS_PTR+53                            ;
                                00C8   CA      D4 000A5          CLRL    INS$G_KFEFAB+24                             ; 1562
    0060  8F       00      6E          00  2C 000A9          MOVC5   #0, (SP), #0, #96, $RMS_PTR                     ; 1572
                                  94   AA      000B0                                                                 ;
                          94   AA  6002    8F  B0 000B2          MOVW    #24578, $RMS_PTR                            ;
                          96   AA          01  8E 000B8          MNEGB   #1, $RMS_PTR+2                              ;
                          98   AA    04    AB  D0 000BC          MOVL    INS$L_KFERSS, $RMS_PTR+4                    ;
                          9C   AA          10  90 000C1          MOVB    #16, $RMS_PTR+8                             ;
                          9E   AA          01  8E 000C5          MNEGB   #1, $RMS_PTR+10                             ;
                          A0   AA          6B  D0 000C9          MOVL    INS$L_KFEESS, $RMS_PTR+12                   ;
                    05    FA   AA          02  E1 000CD          BBC     #2, INS$GL_CTLMSK+2, 4$                     ; 1574
                        00C6   CA          01  90 000D2          MOVB    #1, INS$G_KFEFAB+22                         ; 1575
                                00B0   CA      9F 000D7 4$:      PUSHAB  INS$G_KFEFAB                                ; 1583
                00000000G  00          01  FB 000DB          CALLS   #1, SYS$OPEN                                    ;
                           59          50  D0 000E2          MOVL    R0, OPEN_STATUS                                 ;
                           16          59  E9 000E5          BLBC    OPEN_STATUS, 7$                                 ; 1588
              06      00F1   CA          05  E0 000E8          BBS     #5, DEV+1, 5$                               ; 1594
              07      00F0   CA          05  E1 000EE          BBC     #5, DEV, 6$                                 ;
                           59 00000000G  8F  D0 000F4 5$:      MOVL    #INS$_NOLOAD, OPEN_STATUS                   ; 1596
                           41          59  E8 000FB 6$:      BLBS    OPEN_STATUS, 9$                             ; 1599
                        00C8   CA 007B109A 8F  D0 000FE 7$:      MOVL    #8065178, INS$G_KFEFAB+24                   ; 1602
                                     FC    AB  D5 00107          TSTL    COMMAND_MODE                                ; 1603
                                           39  12 0010A          BNEQ    10$                                         ;
                                     14    AA  9F 0010C          PUSHAB  INS$GQ_FILDSC                               ; 1605
                                   0000'   CF  9F 0010F          PUSHAB  HEL_DSC                                     ;
                00000000G  00              02  FB 00113          CALLS   #2, STR$COMPARE                             ;
                                           50  D5 0011A          TSTL    R0                                          ;
                                           12  13 0011C          BEQL    8$                                          ;
                                     14    AA  9F 0011E          PUSHAB  INS$GQ_FILDSC                               ; 1606
                                   0000'   CF  9F 00121          PUSHAB  HELP_DSC                                    ;
                00000000G  00              02  FB 00125          CALLS   #2, STR$COMPARE                             ;
                                           50  D5 0012C          TSTL    R0                                          ;
                                           15  12 0012E          BNEQ    10$                                         ;
                                00000000G  8F  DD 00130 8$:      PUSHL   #INS$_HELP                                  ; 1609
                00000000G  00              01  FB 00136          CALLS   #1, LIB$SIGNAL                              ;
                                           06  11 0013D          BRB     10$                                         ; 1599
                          F4   AA    00BC   CA  D0 0013F 9$:      MOVL    INS$G_KFEFAB+12, INS$GL_KFFCHAN             ; 1613
                                00B0   CA      9F 00145 10$:     PUSHAB  INS$G_KFEFAB                                ; 1622
                00000000G  00              01  FB 00149          CALLS   #1, UTIL$GETFILENAME                        ;
```

```
                04   AE        60              08  28 00150          MOVC3   #8, (R0), DESC                          ;
                                22              59  E9 00155          BLBC    OPEN_STATUS, 11$                        ; 1623
                                50        CD    AA  9A 00158          MOVZBL  INS$G_KFENAM+57, R0                     ; 1627
                                51        CE    AA  9A 0015C          MOVZBL  INS$G_KFENAM+58, R1                     ;
                                50              51  C0 00160          ADDL2   R1, R0                                  ;
                                51        CF    AA  9A 00163          MOVZBL  INS$G_KFENAM+59, R1                     ; 1628
                                50              51  C0 00167          ADDL2   R1, R0                                  ; 1627
                                51        D0    AA  9A 0016A          MOVZBL  INS$G_KFENAM+60, R1                     ; 1629
                                50              51  C0 0016E          ADDL2   R1, R0                                  ; 1625
                                58        04    AE  3C 00171          MOVZWL  DESC, VERSION_LENGTH                    ;
                                58              50  C2 00175          SUBL2   R0, VERSION_LENGTH                      ;
                                                02  11 00178          BRB     12$                                     ;
                                                58  D4 0017A 11$:      CLRL    VERSION_LENGTH                          ; 1631
                04   BE   04   AE              3A  3A 0017C 12$:      LOCC    #58, DESC, @DESC+4                      ; 1635
                                                02  12 00182          BNEQ    13$                                     ;
                                                51  D4 00184          CLRL    R1                                      ;
                                57              51  D0 00186 13$:      MOVL    R1, PTR                                 ;
                                                59  13 00189          BEQL    14$                                     ; 1636
                     20   AE        24   AE    9E 0018B              MOVAB   DEVNAM, DEV_DSC+4                       ; 1639
                                50   20   AE    D0 00190              MOVL    DEV_DSC+4, R0                           ; 1640
                                60   5F        8F 90 00194          MOVB    #95, (R0)                               ;
                     56         57   08   AE    C3 00198              SUBL3   DESC+4, PTR, R6                         ; 1641
                01   A0   08   BE              56  28 0019D          MOVC3   R6, @DESC+4, 1(R0)                      ; 1642
                1C   AE        56              01  A1 001A3          ADDW3   #1, R6, DEV_DSC                         ; 1643
                     10   AE   04   AA          D0 001A8              MOVL    INS$GQ_KFERNS+4, ITMLST+4               ; 1645
                     14   AE        6A          9E 001AD              MOVAB   INS$GQ_KFERNS, ITMLST+8                 ; 1646
                                                7E  7C 001B1          CLRQ    -(SP)                                   ; 1647
                                                7E  7C 001B3          CLRQ    -(SP)                                   ;
                                1C   AE         9F  001B5             PUSHAB  ITMLST                                  ;
                                30   AE         9F  001B8             PUSHAB  DEV_DSC                                 ;
                                                7E  7C 001BB          CLRQ    -(SP)                                   ;
           00000000G            00              08  FB 001BD          CALLS   #8, SYS$GETDVIW                         ;
                                51        04    AE  3C 001C4          MOVZWL  DESC, R1                                ; 1652
                                51              56  C2 001C8          SUBL2   R6, R1                                  ;
                                50              6A  3C 001CB          MOVZWL  INS$GQ_KFERNS, R0                       ; 1654
                                50        04    AA  C0 001CE          ADDL2   INS$GQ_KFERNS+4, R0                     ;
                60              67              51  28 001D2          MOVC3   R1, (PTR), (R0)                         ; 1653
                                57              53  D0 001D6          MOVL    R3, PTR                                 ;
                51              57        04    AA  C3 001D9          SUBL3   INS$GQ_KFERNS+4, PTR, R1                ; 1655
                6A              51              58  A3 001DE          SUBW3   VERSION_LENGTH, R1, INS$GQ_KFERNS       ; 1656
                                                0C  11 001E2          BRB     15$                                     ; 1636
           04   BA        08   BE        04    AE  28 001E4 14$:      MOVC3   DESC, @DESC+4, @INS$GQ_KFERNS+4         ; 1664
                6A        04   AE              58  A3 001EB          SUBW3   VERSION_LENGTH, DESC, INS$GQ_KFERNS     ; 1665
                                                5A  DD 001F0 15$:      PUSHL   R10                                     ; 1671
                     FD63   CF                01  FB 001F2          CALLS   #1, INS$CVT_DIR                         ;
                                                59  E8 001F7          BLBS    OPEN_STATUS, 17$                        ; 1676
           0001827A 8F   00B8   CA              D1  001FA             CMPL    INS$G_KFEFAB+8, #98938                  ; 1679
                                                14  13 00203          BEQL    16$                                     ;
                                7E   00B8   CA   7D  00205             MOVQ    INS$G_KFEFAB+8, -(SP)                   ; 1682
                                                5A  DD 0020A          PUSHL   R10                                     ; 1681
                                                01  DD 0020C          PUSHL   #1                                      ;
                                     00C8   CA  DD  0020E             PUSHL   INS$G_KFEFAB+24                         ;
           00000000G            00              05  FB 00212          CALLS   #5, LIB$SIGNAL                          ;
                                     08    AA   D4  00219 16$:      CLRL    INS$GL_KFEADR                           ; 1684
                                50              59  D0 0021C          MOVL    OPEN_STATUS, R0                         ; 1685
                                                04  0021F             RET                                             ;
                     08   AA        00C8   CA  D0  00220 17$:      MOVL    INS$G_KFEFAB+24, INS$GL_KFEADR          ; 1692
```

```
                                50          01  D0 00226          MOVL    #1, R0                          ; 1694
                                                04 00229          RET                                     ; 1695

; Routine Size:  554 bytes,    Routine Base:  $CODE$ + 0BB2

; 1515          1696  1
```

```
;  1517          1697  1 %SBTTL  'INS$CLOSE_FILE';
;  1518          1698  1 GLOBAL ROUTINE INS$CLOSE_FILE =
;  1519          1699  2 BEGIN
;  1520          1700  2 !
;  1521          1701  2 !       Be sure to clean up any opened file by deassigning the channel
;  1522          1702  2 !
;  1523          1703  2 IF .INS$GL_KFECHAN NEQ 0
;  1524          1704  2 THEN
;  1525          1705  3     BEGIN
;  1526          1706  3     REPORT ($DASSGN ( CHAN = .INS$GL_KFECHAN ));
;  1527          1707  3     INS$GL_KFECHAN = 0;
;  1528          1708  2     END;
;  1529          1709  2
;  1530          1710  2 RETURN TRUE;
;  1531          1711  1 END;              ! Routine INS$CLOSE_FILE


                                                                .EXTRN  SYS$DASSGN

                                                0000 00000      .ENTRY  INS$CLOSE_FILE, Save nothing                    ; 1698
                              50      0000'  CF  D0 00002       MOVL    INS$GL_KFECHAN, R0                              ; 1703
                                             19  13 00007       BEQL    2$                                              ;
                                             50  DD 00009       PUSHL   R0                                              ; 1706
                          00000000G  00      01  FB 0000B       CALLS   #1, SYS$DASSGN                                  ;
                                     09      50  E8 00012       BLBS    STATUS, 1$                                      ;
                                             50  DD 00015       PUSHL   STATUS                                          ;
                          00000000G  00      01  FB 00017       CALLS   #1, LIB$SIGNAL                                  ;
                                     0000'   CF  D4 0001E 1$:    CLRL    INS$GL_KFECHAN                                  ; 1707
                              50             01  D0 00022 2$:    MOVL    #1, R0                                          ; 1710
                                             04  00025         RET                                                     ; 1711

; Routine Size:  38 bytes,    Routine Base:  $CODE$ + 0DDC

;  1532          1712  1
```

```
;  1534          1713  1 %SBTTL  'INS$CHECK_PRIV';
;  1535          1714  1
;  1536          1715  1 GLOBAL ROUTINE INS$CHECK_PRIV =
;  1537          1716  1 !+++
;  1538          1717  1 !
;  1539          1718  1 !  FUNCTIONAL DESCRIPTION:
;  1540          1719  1 !       Check that user has CMKRNL privilege.
;  1541          1720  1 !
;  1542          1721  1 !  INPUT:
;  1543          1722  1 !       None
;  1544          1723  1 !
;  1545          1724  1 !  OUTPUT:
;  1546          1725  1 !       None
;  1547          1726  1 !
;  1548          1727  1 !  ROUTINE VALUE:
;  1549          1728  1 !       ss$_nocmkrnl    If user does not have CMKRNL privilege
;  1550          1729  1 !       low bit set     If user has CMKRNL privilege
;  1551          1730  1 !---
;  1552          1731  2 BEGIN
;  1553          1732  2 IF NOT .CTL$GQ_PROCPRIV [PRV$V_CMKRNL]
;  1554          1733  2 THEN
;  1555          1734  2     RETURN SS$_NOCMKRNL
;  1556          1735  2 ELSE
;  1557          1736  2     RETURN TRUE;
;  1558          1737  1 END;                  ! routine INS$CHECK_PRIV



                                        0000 00000          .ENTRY  INS$CHECK_PRIV, Save nothing                ; 1715
                    06 00000000G    00  E8 00002          BLBS    CTL$GQ_PROCPRIV, 1$                         ; 1732
                    50      2804    8F  3C 00009          MOVZWL  #10244, R0                                  ; 1736
                                        04 0000E          RET                                                 ;
                    50              01  D0 0000F 1$:      MOVL    #1, R0                                      ;
                                        04 00012          RET                                                 ; 1737

; Routine Size:  19 bytes,    Routine Base:  $CODE$ + 0E02

;  1559          1738  1
```

```
;  1561          1739  1 %SBTTL  'INS_HANDLER';
;  1562          1740  1
;  1563          1741  1 ROUTINE INS_HANDLER (SIGARGS, MECHARGS) =
;  1564          1742  2 BEGIN
;  1565          1743  2 !
;  1566          1744  2 !       This routine is a condition handler called when ever a
;  1567          1745  2 !       SIGNAL is done by INSTALL.  It merely remembers the
;  1568          1746  2 !       most severe error for an exit status.
;  1569          1747  2 !
;  1570          1748  2 MAP
;  1571          1749  2     SIGARGS : REF $BBLOCK,
;  1572          1750  2     MECHARGS : REF $BBLOCK;
;  1573          1751  2
;  1574          1752  2 BIND
;  1575          1753  2     SIGNAME = SIGARGS [CHF$L_SIG_NAME] : $BBLOCK;       !Name of signal
;  1576          1754  2
;  1577          1755  2 IF NOT .SIGNAME AND                                   !If its an error signal
;  1578          1756  4     ((.SIGNAME [STS$V_SEVERITY] GEQU                  ! and severity is worse than it was
;  1579          1757  3         .INS$EXIT_STATUS [STS$V_SEVERITY]) OR
;  1580          1758  3         .INS$EXIT_STATUS [STS$V_SEVERITY])            ! or we haven't had any errors
;  1581          1759  2 THEN
;  1582          1760  2     INS$EXIT_STATUS  = .SIGNAME;                      ! then remember it for exiting
;  1583          1761  2
;  1584          1762  2 RETURN SS$_RESIGNAL
;  1585          1763  1 END;                                  !Of ins_handler


                                        0004 00000 INS_HANDLER:
                                                   .WORD   Save R2                                          ;  1741
                                52      0000'  CF  9E 00002        MOVAB   INS$EXIT_STATUS, R2
                                                                                                            ;
                     50         04      AC      04  C1 00007        ADDL3   #4, SIGARGS, R0                  ;  1753
                                        12      60  E8 0000C        BLBS    (R0), 2$                         ;  1755
           51                   62      03      00  EF 0000F        EXTZV   #0, #3, INS$EXIT_STATUS, R1      ;  1757
           51                   60      03      00  ED 00014        CMPZV   #0, #3, (R0), R1
                                                03  1E 00019        BGEQU   1$                               ;
                                        03      62  E9 0001B        BLBC    INS$EXIT_STATUS, 2$              ;  1758
                                        62      60  D0 0001E 1$:     MOVL    (R0), INS$EXIT_STATUS            ;  1760
                                        50      0918    8F  3C 00021 2$:     MOVZWL  #2328, R0                ;  1762
                                                        04  00026        RET                                  ;  1763

; Routine Size:  39 bytes,    Routine Base:  $CODE$ + 0E15

;  1586          1764  1
;  1587          1765  1 END                                       ! Module INSMAIN
;  1588          1766  0 ELUDOM


                                                   .EXTRN  LIB$SIGNAL

;                                  PSECT SUMMARY
;
```

```
;       Name                    Bytes                           Attributes
;
;  $PLIT$                        539  NOVEC,NOWRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $GLOBAL$                      388  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $OWN$                          60  NOVEC,  WRT,  RD ,NOEXE,NOSHR,  LCL,  REL,  CON,NOPIC,ALIGN(2)
;  $CODE$                       3644  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       130         0      1000          00:01.8




;                                   COMMAND QUALIFIERS
;
;       BLISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS$:INSMAIN/OBJ=OBJ$:INSMAIN MSRC$:INSMAIN/UPDATE=(ENH$:INSMAIN)
;
; Size:         3644 code + 987 data bytes
; Run Time:       00:58.6
; Elapsed Time:   03:19.0
; Lines/CPU Min:   1809
; Lexemes/CPU-Min: 22646
; Memory Used:  308 pages
; Compilation Complete
```

INSMAIN
LIS

INSDELETE
LIS

INSOLDCMD
LIS

INSPRSOLD
LIS

INSGLOBAL
LIS

INSKFSCAN
LIS

INSLIST
LIS

INSMSG
LIS