```
LLL                  IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                  IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                  IIIIIIIII     BBBBBBBBBBBB     RRRRRRRRRRRR     TTTTTTTTTTTTTTT  LLL
LLL                     III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                     III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                     III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                     III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                     III        BBB         BBB  RRR         RRR        TTT        LLL
LLL                     III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                     III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                     III        BBBBBBBBBBBB     RRRRRRRRRRRR           TTT        LLL
LLL                     III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                     III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                     III        BBB         BBB  RRR   RRR              TTT        LLL
LLL                     III        BBB         BBB  RRR      RRR           TTT        LLL
LLL                     III        BBB         BBB  RRR      RRR           TTT        LLL
LLL                     III        BBB         BBB  RRR      RRR           TTT        LLL
LLLLLLLLLLLLLLL      IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL      IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
LLLLLLLLLLLLLLL      IIIIIIIII     BBBBBBBBBBBB     RRR         RRR        TTT        LLLLLLLLLLLLLLL
```

FILEID**LIBDELETE

```
LL              IIIIII    BBBBBBBB    DDDDDDDD    EEEEEEEEEE  LL          EEEEEEEEEE  TTTTTTTTTT  EEEEEEEEEE
LL              IIIIII    BBBBBBBB    DDDDDDDD    EEEEEEEEEE  LL          EEEEEEEEEE  TTTTTTTTTT  EEEEEEEEEE
LL                II      BB      BB  DD      DD  EE          LL          EE              TT      EE
LL                II      BB      BB  DD      DD  EE          LL          EE              TT      EE
LL                II      BB      BB  DD      DD  EE          LL          EE              TT      EE
LL                II      BB      BB  DD      DD  EE          LL          EE              TT      EE
LL                II      BBBBBBBB    DD      DD  EEEEEEEE    LL          EEEEEEEE        TT      EEEEEEEE
LL                II      BBBBBBBB    DD      DD  EEEEEEEE    LL          EEEEEEEE        TT      EEEEEEEE
LL                II      BB      BB  DD      DD  EE          LL          EE              TT      EE
LL                II      BB      BB  DD      DD  EE          LL          EE              TT      EE
LL                II      BB      BB  DD      DD  EE          LL          EE              TT      EE
LL                II      BB      BB  DD      DD  EE          LL          EE              TT      EE
LLLLLLLLLL      IIIIII    BBBBBBBB    DDDDDDDD    EEEEEEEEEE  LLLLLLLLLL  EEEEEEEEEE      TT      EEEEEEEEEE
LLLLLLLLLL      IIIIII    BBBBBBBB    DDDDDDDD    EEEEEEEEEE  LLLLLLLLLL  EEEEEEEEEE      TT      EEEEEEEEEE


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

```
;     1          0001  0 MODULE LIB$DELETE_FILE ( %TITLE 'Delete one or more files'
;     2          0002  0                    IDENT = '1-006'          ! File: LIBDELETE.B32 Edit: BLS0331
;     3          0003  0                    ) =
;     4          0004  1 BEGIN
;     5          0005  1 !
;     6          0006  1 !**************************************************************************
;     7          0007  1 !*                                                                        *
;     8          0008  1 !*  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY                               *
;     9          0009  1 !*  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.                 *
;    10          0010  1 !*  ALL RIGHTS RESERVED.                                                   *
;    11          0011  1 !*                                                                        *
;    12          0012  1 !*  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED   *
;    13          0013  1 !*  ONLY IN  ACCORDANCE WITH  THE  TERMS  OF  SUCH  LICENSE  AND WITH THE  *
;    14          0014  1 !*  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR  ANY  OTHER   *
;    15          0015  1 !*  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY   *
;    16          0016  1 !*  OTHER PERSON.  NO TITLE TO AND OWNERSHIP OF  THE  SOFTWARE IS  HEREBY   *
;    17          0017  1 !*  TRANSFERRED.                                                           *
;    18          0018  1 !*                                                                        *
;    19          0019  1 !*  THE INFORMATION IN THIS SOFTWARE IS  SUBJECT TO CHANGE WITHOUT NOTICE   *
;    20          0020  1 !*  AND  SHOULD  NOT  BE  CONSTRUED AS  A COMMITMENT BY DIGITAL EQUIPMENT   *
;    21          0021  1 !*  CORPORATION.                                                           *
;    22          0022  1 !*                                                                        *
;    23          0023  1 !*  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE  OR  RELIABILITY OF ITS  *
;    24          0024  1 !*  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.                *
;    25          0025  1 !*                                                                        *
;    26          0026  1 !*                                                                        *
;    27          0027  1 !**************************************************************************
;    28          0028  1 !
;    29          0029  1
;    30          0030  1 !++
;    31          0031  1 ! FACILITY:     General Utility Library
;    32          0032  1 !
;    33          0033  1 ! ABSTRACT:
;    34          0034  1 !
;    35          0035  1 !       LIB$DELETE_FILE deletes one or more files.
;    36          0036  1 !
;    37          0037  1 ! ENVIRONMENT:  User mode - AST reentrant
;    38          0038  1 !
;    39          0039  1 ! AUTHOR: Steven B. Lionel, CREATION DATE: 13-July-1982
;    40          0040  1 !
;    41          0041  1 ! MODIFIED BY:
;    42          0042  1 !
;    43          0043  1 ! 1-001 - Original.  SBL 13-July-1982
;    44          0044  1 ! 1-002 - Add related-file-spec argument, and pass FAB to confirm-routine.
;    45          0045  1 !         SBL 1-Oct-1982
;    46          0046  1 ! 1-003 - Pass error source code to error-routine.  SBL 19-Nov-1982
;    47          0047  1 ! 1-004 - Update for new argument to LIB$FILE_SCAN. BLS  6-FEB-1984
;    48          0048  1 ! 1-005 - Add new argument for filescan context. BLS  5-MAR-1984
;    49          0049  1 ! 1-006 - Parse null string to deallocate rms context after calling
;    50          0050  1 !       file_scan. BLS  9-JUL-1984
;    51          0051  1 !--
;    52          0052  1
```

```
;    54          0053  1 %SBTTL 'Declarations'
;    55          0054  1 !
;    56          0055  1 ! PROLOGUE FILE:
;    57          0056  1 !
;    58          0057  1
;    59          0058  1 REQUIRE 'RTLIN:LIBPROLOG';                  ! Switches, PSECTS, macros, etc.
;    60          0129  1
;    61          0130  1 !
;    62          0131  1 ! LINKAGES:
;    63          0132  1 !
;    64          0133  1 !       NONE
;    65          0134  1 !
;    66          0135  1 ! TABLE OF CONTENTS:
;    67          0136  1 !
;    68          0137  1
;    69          0138  1 FORWARD ROUTINE
;    70          0139  1     LIB$DELETE_FILE,                       ! Delete one or more files
;    71          0140  1     DO_DELETE: NOVALUE,                    ! Delete a file
;    72          0141  1     DELETE_ERROR: NOVALUE,                 ! Error routine
;    73          0142  1     DELETE_HANDLER;                        ! Condition handler
;    74          0143  1
;    75          0144  1 !
;    76          0145  1 ! MACROS:
;    77          0146  1 !
;    78          0147  1 !       NONE
;    79          0148  1 !
;    80          0149  1 ! EQUATED SYMBOLS:
;    81          0150  1 !
;    82          0151  1 !       NONE
;    83          0152  1 !
;    84          0153  1 ! FIELDS:
;    85          0154  1 !
;    86          0155  1 !       NONE
;    87          0156  1 !
;    88          0157  1 ! OWN STORAGE:
;    89          0158  1 !
;    90          0159  1 !       NONE
;    91          0160  1 !
;    92          0161  1 ! EXTERNALS:
;    93          0162  1 !
;    94          0163  1
;    95          0164  1 EXTERNAL ROUTINE
;    96          0165  1     LIB$ANALYZE_SDESC_R2: LIB$ANALYZE_SDESC_R2$LINKAGE,
;    97          0166  1     LIB$FILE_SCAN,
;    98          0167  1     LIB$SCOPY_DXDX,
;    99          0168  1     LIB$SCOPY_R_DX,
;   100          0169  1     LIB$SIG_TO_RET;
;   101          0170  1
;   102          0171  1 EXTERNAL LITERAL
;   103          0172  1     LIB$_ERRROUCAL,
;   104          0173  1     LIB$_INVFILSPE;
```

```
;  106          0174  1 %SBTTL 'LIB$DELETE_FILE - Delete one or more files'
;  107          0175  1 GLOBAL ROUTINE LIB$DELETE_FILE (
;  108          0176  1     FILESPEC: REF BLOCK [, BYTE],                  ! File to delete
;  109          0177  1     DEFAULT_FILESPEC: REF BLOCK [, BYTE],          ! Default file specification
;  110          0178  1     RELATED_FILESPEC: REF BLOCK [, BYTE],          ! Related file specification
;  111          0179  1     SUCCESS_ROUTINE,                               ! Called on successful delet
;  112          0180  1     ERROR_ROUTINE,                                 ! Called on error
;  113          0181  1     CONFIRM_ROUTINE,                               ! Called for confirmation
;  114          0182  1     USER_ARG,                                      ! User argument
;  115          0183  1     RESULTANT_NAME: REF BLOCK [, BYTE],            ! Returned filespec
;  116          0184  1     FILE_SCAN_CONTEXT                              ! Lib$file_scan context
;  117          0185  1     ) =
;  118          0186  1
;  119          0187  1 !++
;  120          0188  1 ! FUNCTIONAL DESCRIPTION:
;  121          0189  1 !
;  122          0190  1 !       LIB$DELETE_FILE deletes one or more files.  It is similar in
;  123          0191  1 !       function to the DCL DELETE command.  The specification
;  124          0192  1 !       of the file(s) to be deleted may include wild cards.
;  125          0193  1 !
;  126          0194  1 !
;  127          0195  1 ! CALLING SEQUENCE:
;  128          0196  1 !
;  129          0197  1 !       ret_status.wlc.v = LIB$DELETE_FILE (
;  130          0198  1 !               FILESPEC.rt.dx,
;  131          0199  1 !               [, [DEFAULT_FILESPEC.rt.dx]
;  132          0200  1 !               [, [RELATED_FILESPEC.rt.dx]
;  133          0201  1 !               [, [SUCCESS_ROUTINE.szem.r]
;  134          0202  1 !               [, [ERROR_ROUTINE.fzemlc.r]
;  135          0203  1 !               [, [CONFIRM_ROUTINE.fzemlc.r]
;  136          0204  1 !               [, [USER_ARG.rz]
;  137          0205  1 !               [, [RESULTANT_NAME.wt.dx]
;  138          0206  1 !               [, [FILE_SCAN_CONTEXT] ]]]]]]])
;  139          0207  1 !
;  140          0208  1 ! FORMAL PARAMETERS:
;  141          0209  1 !
;  142          0210  1 !       FILESPEC                - The file specification of the file(s)
;  143          0211  1 !                                 to be deleted.  Passed by descriptor.
;  144          0212  1 !                                 The specification may include wild cards.
;  145          0213  1 !
;  146          0214  1 !       DEFAULT_FILESPEC        - The default file specification of the
;  147          0215  1 !                                 file(s) to be deleted.  Passed by descriptor.
;  148          0216  1 !                                 If omitted, the default is the null string.
;  149          0217  1 !
;  150          0218  1 !       RELATED_FILESPEC        - The related file specification of the
;  151          0219  1 !                                 files to be deleted.  Passed by descriptor.
;  152          0220  1 !                                 If omitted, the default is the null string.
;  153          0221  1 !                                 "Input file parsing" is used.
;  154          0222  1 !
;  155          0223  1 !       SUCCESS_ROUTINE         - The entry mask of a routine to call for
;  156          0224  1 !                                 each successful delete, passed by reference.
;  157          0225  1 !                                 The calling format of the SUCCESS_ROUTINE is
;  158          0226  1 !                                 as follows:
;  159          0227  1 !                                       CALL SUCCESS_ROUTINE (
;  160          0228  1 !                                           filespec.rt.ds,
;  161          0229  1 !                                           user_arg.rz)
;  162          0230  1 !
```

```
!                       filespec     - The RMS resultant file specification
!                                      of the file being deleted.  If
!                                      RESULTANT_NAME was specified,
!                                      it is used to pass the string
!                                      to SUCCESS_ROUTINE.  Otherwise,
!                                      a class S, type T string is passed.
!
!
!                       user_arg     - The value of user_arg passed
!                                      to LIB$DELETE_FILE is passed
!                                      to SUCCESS_ROUTINE using the
!                                      same mechanism as was used to
!                                      pass it to LIB$DELETE_FILE
!
!
!       ERROR_ROUTINE           - The entry mask of a routine to call when
!                                 an error is detected, passed by reference.
!                                 The function value returned by the routine
!                                 determines whether or not more files will
!                                 be processed.
!
!                                 The calling format of the ERROR_ROUTINE is
!                                 as follows:
!                                       ret_status.wlc.v = ERROR_ROUTINE (
!                                           filespec.rt.ds,
!                                           RMS_sts.rlc.r
!                                           RMS_stv.rlc.r,
!                                           error_source.rl.r,
!                                           user_arg.rz)
!
!                       filespec     - The RMS resultant file specification
!                                      of the file being deleted.  If
!                                      RESULTANT_NAME was specified,
!                                      it is used to pass the string
!                                      to ERROR_ROUTINE.  Otherwise,
!                                      a class S, type T string is passed.
!
!                       RMS_sts      - The primary condition code
!                                      which describes the error
!                                      that occurred.
!
!                       RMS_stv      - The secondary condition code
!                                      which describes the error
!                                      that occurred.
!
!                       error_source - An integer code that indicates
!                                      at what point the error was
!                                      found.  The values are:
!                                        0 = Error searching for
!                                              filespec
!                                        1 = Error deleting file
!
!                       user_arg     - The value of user_arg passed
!                                      to LIB$DELETE_FILE is passed
!                                      to ERROR_ROUTINE using the
!                                      same mechanism as was used to
!                                      pass it to LIB$DELETE_FILE
```

```
1 !                                                If ERROR_ROUTINE returns a success status,
1 !                                                then processing of files will continue.  If
1 !                                                a failure status is returned, then processing
1 !                                                will cease immediately and LIB$DELETE_FILE
1 !                                                will return with the error status.
1 !
1 !                                                If ERROR_ROUTINE is not specified,
1 !                                                LIB$DELETE_FILE will return to its caller with
1 !                                                the most severe of the error statuses encountered
1 !                                                while deleting the files.  Otherwise, if
1 !                                                ERROR_ROUTINE is called for an error, the
1 !                                                success status LIB$_ERRROUCAL is returned.
1 !                                                Note that ERROR_ROUTINE is not called for
1 !                                                errors related to string copying.
1 !
1 !               CONFIRM_ROUTINE              - The entry mask of a routine to call before
1 !                                                each file is deleted, passed by reference.
1 !                                                The function value returned by the routine
1 !                                                determines whether or not the file will be
1 !                                                be deleted.
1 !
1 !                                                The calling format of CONFIRM_ROUTINE is
1 !                                                as follows:
1 !                                                     ret_status.wlc.v = CONFIRM_ROUTINE (
1 !                                                         filespec.rt.ds,
1 !                                                         FAB.rr.r,
1 !                                                         user_arg.rz)
1 !
1 !                                                  filespec     - The RMS resultant file specification
1 !                                                                 of the file being deleted.  If
1 !                                                                 RESULTANT_NAME was specified,
1 !                                                                 it is used to pass the string
1 !                                                                 to CONFIRM_ROUTINE.  Otherwise,
1 !                                                                 a class S, type T string is passed.
1 !
1 !                                                  FAB          - The address of the RMS FAB
1 !                                                                 that describes the file
1 !                                                                 being deleted.  You may
1 !                                                                 perform an RMS $OPEN on the
1 !                                                                 FAB to obtain file attributes
1 !                                                                 you need to determine whether
1 !                                                                 the file should be deleted,
1 !                                                                 but you must close the file
1 !                                                                 with $CLOSE before returning
1 !                                                                 to LIB$DELETE_FILE.
1 !
1 !                                                  user_arg     - The value of user_arg passed
1 !                                                                 to LIB$DELETE_FILE is passed
1 !                                                                 to CONFIRM_ROUTINE using the
1 !                                                                 same mechanism as was used to
1 !                                                                 pass it to LIB$DELETE_FILE
1 !
1 !                                                If CONFIRM_ROUTINE returns success, the file
1 !                                                is then deleted, otherwise that file is not
1 !                                                deleted.
1 !
```

```
1 !        USER_ARG                      - A value passed to SUCCESS_ROUTINE,
1 !                                        ERROR_ROUTINE and CONFIRM_ROUTINE each
1 !                                        time they are called.  Whatever mechanism
1 !                                        is used to pass USER_ARG to LIB$DELETE_FILE is
1 !                                        used to pass it to the action routine.  This
1 !                                        is an optional parameter, if omitted, zero
1 !                                        is passed by immediate value.
1 !
1 !        RESULTANT_NAME                - A string into which is written the RMS resultant
1 !                                        file specification of the last file processed
1 !                                        by LIB$DELETE_FILE.  Passed by descriptor.
1 !                                        If present, RESULTANT_NAME is used to store
1 !                                        the filespec passed to the action routines
1 !                                        instead of a default class S, type T string.
1 !
1 !        FILE_SCAN_CONTEXT             - The address of a longword, which is
1 !                                        initialized to 0 before calling LIB$DELETE_FILE.
1 !                                        This context is used by LIB$FILE_SCAN to retain
1 !                                        multiple input file related file context, and
1 !                                        need only be specified if dealing with multiple
1 !                                        input files, as the DCL DELETE command does.
1 !                                        The context allocated by LIB$FILE_SCAN while
1 !                                        processing the LIB$DELETE_FILE requests may
1 !                                        be deallocated by calling LIB$FILE_SCAN_END
1 !                                        after all calls to LIB$DELETE_FILE have been
1 !                                        completed.
1 !
1 ! IMPLICIT INPUTS:
1 !
1 !        NONE
1 !
1 ! IMPLICIT OUTPUTS:
1 !
1 !        NONE
1 !
1 ! COMPLETION STATUS:
1 !
1 !        SS$_NORMAL       Normal successful completion
1 !        LIB$_ERRROUCAL   Success - error routine called.  A file error was
1 !                         encountered but ERROR_ROUTINE was called to handle
1 !                         the condition.
1 !        LIB$_INVSTRDES   Invalid string descriptor
1 !        LIB$_INVFILSPE   Invalid file specification.  FILESPEC or
1 !                         DEFAULT_FILESPEC is longer than 255 characters.
1 !        LIB$_WRONUMARG   Wrong number of arguments.
1 !        LIB$_xxx         Any error status from LIB$SCOPY_xxx
1 !        RMS$_xxx         Any error status from RMS.  If ERROR_ROUTINE is not
1 !                         specified, this is the most severe of the RMS errors
1 !                         encountered while deleting the file(s).
1 !
1 ! SIDE EFFECTS:
1 !
1 !        The files specified are deleted.
1 !
1 !--
1
2    BEGIN
```

```
;  334          0402  2
;  335          0403  2
;  336          0404  2      LOCAL
;  337          0405  2          FAB: $FAB_DECL,                         ! File's FAB
;  338          0406  2          NAM: $NAM_DECL,                         ! File's NAM
;  339          0407  2          RLF_NAM: $NAM_DECL,                     ! Related file NAM block
;  340          0408  2          ESN: VECTOR [NAM$C_MAXRSS, BYTE],       ! Expanded name string
;  341          0409  2          RSN: VECTOR [NAM$C_MAXRSS, BYTE],       ! Resultant name string
;  342          0410  2          ERROR_SOURCE,                           ! Reason for the error, passed
;  343          0411  2                                                  ! to error_routine.  Values are:
;  344          0412  2                                                  !    0 - error from search
;  345          0413  2                                                  !    1 - error from $ERASE
;  346          0414  2          WORST_ERROR,                            ! Worst error so far
;  347          0415  2          INTERCEPT_FLAG: VOLATILE,               ! 1 if signals are to be intercepted
;  348          0416  2          RET_STATUS;                             ! Return status
;  349          0417  2
;  350          0418  2      BUILTIN
;  351          0419  2          ACTUALCOUNT,
;  352          0420  2          NULLPARAMETER;
;  353          0421  2
;  354          0422  2      !+
;  355          0423  2      ! Enable DELETE_HANDLER as a handler so that it will convert our
;  356          0424  2      ! signals to return statuses if INTERCEPT_FLAG is set.
;  357          0425  2      !-
;  358          0426  2
;  359          0427  2      ENABLE
;  360          0428  2          DELETE_HANDLER (INTERCEPT_FLAG);
;  361          0429  2
;  362          0430  2      !+
;  363          0431  2      ! Validate the argument count.
;  364          0432  2      !-
;  365          0433  2
;  366          0434  2      $LIB$VALIDATE_ARGCOUNT (1,9);
;  367          0435  2
;  368          0436  2      !+
;  369          0437  2      ! Initialize the FAB and NAM blocks.
;  370          0438  2      !-
;  371          0439  2
;  372          0440  2      $FAB_INIT (FAB=FAB, NAM=NAM);
;  373        P 0441  2      $NAM_INIT (NAM=NAM, ESA=ESN, ESS=NAM$C_MAXRSS,
;  374          0442  2          RSA=RSN, RSS=NAM$C_MAXRSS, RLF=RLF_NAM);
;  375          0443  2      $NAM_INIT (NAM=RLF_NAM);
;  376          0444  2
;  377          0445  2      !+
;  378          0446  2      ! Initialize WORST_ERROR to zero.  If any errors occur, the worst one
;  379          0447  2      ! will be in WORST_ERROR when LIB$FILE_SCAN returns.
;  380          0448  2      ! Initialize ERROR_SOURCE to zero.  If DO_DELETE calls DELETE_ERROR
;  381          0449  2      ! it will set ERROR_SOURCE to 1.
;  382          0450  2      !-
;  383          0451  2
;  384          0452  2      WORST_ERROR = 0;
;  385          0453  2      ERROR_SOURCE = 0;
;  386          0454  2
;  387          0455  2      !+
;  388          0456  2      ! Set up FAB for the file specification.  Return LIB$_INVFILSPE
;  389          0457  2      ! if the string is longer than 255 characters.
;  390          0458  2      !-
```

```
;   391     0459  2
;   392     0460  3          BEGIN
;   393     0461  3
;   394     0462  3          LOCAL
;   395     0463  3              FILESPEC_LENGTH: WORD,      ! Length of FILESPEC string
;   396     0464  3              STATUS;                     ! Status from LIB$ANALYZE_SDESC.
;   397     0465  3
;   398     0466  3          STATUS = LIB$ANALYZE_SDESC_R2 (FILESPEC [0,0,0,0];
;   399     0467  3              FILESPEC_LENGTH, FAB [FAB$L_FNA]);
;   400     0468  3
;   401     0469  3          IF NOT .STATUS
;   402     0470  3          THEN
;   403     0471  3              RETURN .STATUS;
;   404     0472  3
;   405     0473  3          IF .FILESPEC_LENGTH GTRU 255
;   406     0474  3          THEN
;   407     0475  3              RETURN LIB$_INVFILSPE;
;   408     0476  3
;   409     0477  3          FAB [FAB$B_FNS] = .FILESPEC_LENGTH;
;   410     0478  2          END;
;   411     0479  2
;   412     0480  2      !+
;   413     0481  2      ! Get DEFAULT_FILESPEC if present, and set up as default name string.
;   414     0482  2      !-
;   415     0483  2
;   416     0484  2      IF NOT NULLPARAMETER (2)
;   417     0485  2      THEN
;   418     0486  3          BEGIN
;   419     0487  3
;   420     0488  3          LOCAL
;   421     0489  3              FILESPEC_LENGTH: WORD,      ! Length of DEFAULT_FILESPEC string
;   422     0490  3              STATUS;                     ! Status from LIB$ANALYZE_SDESC.
;   423     0491  3
;   424     0492  3          STATUS = LIB$ANALYZE_SDESC_R2 (DEFAULT_FILESPEC [0,0,0,0];
;   425     0493  3              FILESPEC_LENGTH, FAB [FAB$L_DNA]);
;   426     0494  3
;   427     0495  3          IF NOT .STATUS
;   428     0496  3          THEN
;   429     0497  3              RETURN .STATUS;
;   430     0498  3
;   431     0499  3          IF .FILESPEC_LENGTH GTRU 255
;   432     0500  3          THEN
;   433     0501  3              RETURN LIB$_INVFILSPE;
;   434     0502  3
;   435     0503  3          FAB [FAB$B_DNS] = .FILESPEC_LENGTH;
;   436     0504  2          END;
;   437     0505  2
;   438     0506  2      !+
;   439     0507  2      ! Get RELATED_FILESPEC if present, and set up as related name string.
;   440     0508  2      !-
;   441     0509  2
;   442     0510  2      IF NOT NULLPARAMETER (3)
;   443     0511  2          AND NULLPARAMETER (9)
;   444     0512  2      THEN
;   445     0513  3          BEGIN
;   446     0514  3
;   447     0515  3          LOCAL
```

```
3               FILESPEC_LENGTH: WORD,      ! Length of RELATED_FILESPEC string
3               STATUS;                     ! Status from LIB$ANALYZE_SDESC.
3
3           STATUS = LIB$ANALYZE_SDESC_R2 (RELATED_FILESPEC [0,0,0,0];
3               FILESPEC_LENGTH, RLF_NAM [NAM$L_RSA]);
3
3           IF NOT .STATUS
3           THEN
3               RETURN .STATUS;
3
3           IF .FILESPEC_LENGTH GTRU 255
3           THEN
3               RETURN LIB$_INVFILSPE;
3
3           RLF_NAM [NAM$B_RSL] = .FILESPEC_LENGTH;
2           END;
2
2       !+
2       ! Call LIB$FILE_SCAN, which will call DO_DELETE for each file that is
2       ! to be deleted.  Pass as extra parameters the success routine, error routine,
2       ! confirm routine, user argument, result filename descriptor, worst error
2       ! holder, error_source and intercept flag.  These will be passed to DO_DELETE by
2       ! LIB$FILE_SCAN.
2       !-
2
2       LIB$FILE_SCAN (
2           FAB,                            ! Input FAB
2           DO_DELETE,                      ! Success routine
2           DELETE_ERROR,                   ! Error routine
2           (IF ACTUALCOUNT () GEQU 9 THEN .FILE_SCAN_CONTEXT ELSE 0),! context
2           (IF ACTUALCOUNT () GEQU 4 THEN .SUCCESS_ROUTINE ELSE 0),! Success routine
2           (IF ACTUALCOUNT () GEQU 5 THEN .ERROR_ROUTINE ELSE 0),  ! Error routine
2           (IF ACTUALCOUNT () GEQU 6 THEN .CONFIRM_ROUTINE ELSE 0),! Confirm routine
2           (IF ACTUALCOUNT () GEQU 7 THEN .USER_ARG ELSE 0),       ! User argument
2           (IF ACTUALCOUNT () GEQU 8 THEN .RESULTANT_NAME ELSE 0), ! Resultant name
2           WORST_ERROR,                    ! Worst error so far
2           ERROR_SOURCE,                   ! Code for source of error
2           INTERCEPT_FLAG);                ! Signal intercept flag
2
2       !+
2       ! Parse the null string to deallocate any internal
2       ! RMS context.
2       !-
2
2       NAM[NAM$V_SVCTX] = 0;
2       NAM[NAM$V_SYNCHK] = 1;
2       NAM[NAM$B_ESL] = 0;
2       NAM[NAM$B_RSL] = 0;
2       NAM[NAM$B_ESS] = 0;
2       NAM[NAM$B_RSS] = 0;
2       NAM[NAM$L_RLF] = 0;
2       FAB[FAB$B_FNS] = 0;
2       FAB[FAB$B_DNS] = 0;
2       $PARSE(FAB=FAB);
2
2       !+
2       ! Return WORST_ERROR or SS$_NORMAL, as appropriate.
```

```
;       2       !-
;       2
;       2       IF .WORST_ERROR NEQ 0
;       2       THEN
;       2           RET_STATUS = .WORST_ERROR
;       2       ELSE
;       2           RET_STATUS = SS$_NORMAL;
;       2
;       2       RETURN .RET_STATUS;
;       2
;       1       END;                                    ! End of routine LIB$DELETE_FILE

                                                  .TITLE  LIB$DELETE_FILE Delete one or more files
                                                  .IDENT  \1-006\

                                                  .EXTRN  LIB$ANALYZE_SDESC_R2
                                                  .EXTRN  LIB$FILE_SCAN, LIB$SCOPY_DXDX
                                                  .EXTRN  LIB$SCOPY_R_DX, LIB$SIG_TO_RET
                                                  .EXTRN  LIB$_ERRROUCAL, LIB$_INVFILSPE
                                                  .EXTRN  LIB$_WRONUMARG, SYS$PARSE

                                                  .PSECT  _LIB$CODE,NOWRT,  SHR,  PIC,2

                                   007C 00000     .ENTRY  LIB$DELETE_FILE, Save R2,R3,R4,R5,R6          ; 0175
                 56 00000000G   00   9E 00002     MOVAB   LIB$ANALYZE_SDESC_R2, R6                      ;
                 5E     FCE4    CE   9E 00009     MOVAB   -796(SP), SP                                  ;
                          08    AE   D4 0000E     CLRL    INTERCEPT_FLAG                                ; 0401
                 6D     0176    CF   DE 00011     MOVAL   22$, (FP)                                     ;
         50      6C             01   83 00016     SUBB3   #1, (AP), DIFF                                ; 0434
                 08             50   91 0001A     CMPB    DIFF, #8                                      ;
                                08   1B 0001D     BLEQU   1$                                            ;
                 50 00000000G   8F   D0 0001F     MOVL    #LIB$_WRONUMARG, R0                           ;
                                     04 00026     RET                                                   ;
0050  8F         00      6E     00   2C 00027 1$: MOVC5   #0, (SP), #0, #80, $RMS_PTR                   ; 0440
                          B0    AD      0002E                                                           ;
            B0   AD     5003    8F   B0 00030     MOVW    #20483, $RMS_PTR                              ;
            C6   AD             02   90 00036     MOVB    #2, $RMS_PTR+22                               ;
            CF   AD             02   90 0003A     MOVB    #2, $RMS_PTR+31                               ;
            D8   AD     FF50    CD   9E 0003E     MOVAB   NAM, $RMS_PTR+40                              ;
0060  8F         00      6E     00   2C 00044     MOVC5   #0, (SP), #0, #96, $RMS_PTR                   ; 0442
                        FF50    CD      0004B                                                           ;
          FF50   CD     6002    8F   B0 0004E     MOVW    #24578, $RMS_PTR                              ;
          FF52   CD             01   8E 00055     MNEGB   #1, $RMS_PTR+2                                ;
          FF54   CD       0C    AE   9E 0005A     MOVAB   RSN, $RMS_PTR+4                               ;
          FF5A   CD             01   8E 00060     MNEGB   #1, $RMS_PTR+10                               ;
          FF5C   CD     010C    CE   9E 00065     MOVAB   ES'., $RMS_PTR+12                             ;
          FF60   CD     FEF0    CD   9E 0006C     MOVAB   RLF_NAM, $RMS_PTR+16                          ;
0060  8F         00      6E     00   2C 00073     MOVC5   #0, (SP), #0, #96, $RMS_PTR                   ; 0443
                        FEF0    CD      0007A                                                           ;
          FEF0   CD     6002    8F   B0 0007D     MOVW    #24578, $RMS_PTR                              ;
                                6E   7C 00084     CLRQ    ERROR_SOURCE                                  ; 0453
                 50       04    AC   D0 00086     MOVL    FILESPEC, R0                                  ; 0466
                                66   16 0008A     JSB     LIB$ANALYZE_SDESC_R2                          ;
            DC   AD             52   D0 0008C     MOVL    R2, FAB+44                                    ; 0467
                 4C             50   E9 00090     BLBC    STATUS, 4$                                    ; 0469
          00FF   8F             51   B1 00093     CMPW    FILESPEC_LENGTH, #255                         ; 0473
```

```
                    50  1A 00098       BGTRU   6$                                ;
      E4  AD        51  90 0009A       MOVB    FILESPEC_LENGTH, FAB+52           ; 0477
          02        6C  91 0009E       CMPB    (AP), #2                          ; 0484
                    1D  1F 000A1       BLSSU   2$                                ;
              08    AC  D5 000A3       TSTL    8(AP)                             ;
                    18  13 000A6       BEQL    2$                                ;
          50  08    AC  D0 000A8       MOVL    DEFAULT_FILESPEC, R0              ; 0492
                    66  16 000AC       JSB     LIB$ANALYZE_SDESC_R2              ;
      E0  AD        52  D0 000AE       MOVL    R2, FAB+48                        ; 0493
          2A        50  E9 000B2       BLBC    STATUS, 4$                        ; 0495
    00FF  8F        51  B1 000B5       CMPW    FILESPEC_LENGTH, #255             ; 0499
                    2E  1A 000BA       BGTRU   6$                                ;
      E5  AD        51  90 000BC       MOVB    FILESPEC_LENGTH, FAB+53           ; 0503
          03        6C  91 000C0 2$:   CMPB    (AP), #3                          ; 0510
                    32  1F 000C3       BLSSU   8$                                ;
              0C    AC  D5 000C5       TSTL    12(AP)                            ;
                    2D  13 000C8       BEQL    8$                                ;
          09        6C  91 000CA       CMPB    (AP), #9                          ; 0511
                    05  1F 000CD       BLSSU   3$                                ;
              24    AC  D5 000CF       TSTL    36(AP)                            ;
                    23  12 000D2       BNEQ    8$                                ;
          50  0C    AC  D0 000D4 3$:   MOVL    RELATED_FILESPEC, R0              ; 0519
                    66  16 000D8       JSB     LIB$ANALYZE_SDESC_R2              ;
    FEF4  CD        52  D0 000DA       MOVL    R2, RLF_NAM+4                     ; 0520
          01        50  E8 000DF 4$:   BLBS    STATUS, 5$                        ; 0522
                    04     000E2       RET                                       ;
    00FF  8F        51  B1 000E3 5$:   CMPW    FILESPEC_LENGTH, #255             ; 0526
                    08  1B 000E8       BLEQU   7$                                ;
          50 00000000G  8F  D0 000EA 6$:   MOVL    #LIB$_INVFILSPE, R0           ; 0528
                    04     000F1       RET                                       ;
    FEF3  CD        51  90 000F2 7$:   MOVB    FILESPEC_LENGTH, RLF_NAM+3        ; 0530
              08    AE  9F 000F7 8$:   PUSHAB  INTERCEPT_FLAG                    ; 0541
              04    AE  9F 000FA       PUSHAB  ERROR_SOURCE                      ;
              0C    AE  9F 000FD       PUSHAB  WORST_ERROR                       ;
          08        6C  91 00100       CMPB    (AP), #8                          ; 0550
                    05  1F 00103       BLSSU   9$                                ;
              20    AC  DD 00105       PUSHL   RESULTANT_NAME                    ;
                    02  11 00108       BRB     10$                               ;
                    7E  D4 0010A 9$:   CLRL    -(SP)                             ;
          07        6C  91 0010C 10$:  CMPB    (AP), #7                          ; 0549
                    05  1F 0010F       BLSSU   11$                               ;
              1C    AC  DD 00111       PUSHL   USER_ARG                          ;
                    02  11 00114       BRB     12$                               ;
                    7E  D4 00116 11$:  CLRL    -(SP)                             ;
          06        6C  91 00118 12$:  CMPB    (AP), #6                          ; 0548
                    05  1F 0011B       BLSSU   13$                               ;
              18    AC  DD 0011D       PUSHL   CONFIRM_ROUTINE                   ;
                    02  11 00120       BRB     14$                               ;
                    7E  D4 00122 13$:  CLRL    -(SP)                             ;
          05        6C  91 00124 14$:  CMPB    (AP), #5                          ; 0547
                    05  1F 00127       BLSSU   15$                               ;
              14    AC  DD 00129       PUSHL   ERROR_ROUTINE                     ;
                    02  11 0012C       BRB     16$                               ;
                    7E  D4 0012E 15$:  CLRL    -(SP)                             ;
          04        6C  91 00130 16$:  CMPB    (AP), #4                          ; 0546
                    05  1F 00133       BLSSU   17$                               ;
              10    AC  DD 00135       PUSHL   SUCCESS_ROUTINE                   ;
```

```
                                 02  11 00138          BRB     18$
                                 7E  D4 0013A 17$:     CLRL    -(SP)
                           09    6C  91 0013C 18$:     CMPB    (AP), #9                                        ; 0545
                                 05  1F 0013F          BLSSU   19$
                             24  AC  DD 00141          PUSHL   FILE_SCAN_CONTEXT
                                 02  11 00144          BRB     20$
                                 7E  D4 00146 19$:     CLRL    -(SP)
                         0000V   CF  9F 00148 20$:     PUSHAB  DELETE_ERROR                                    ; 0541
                         0000V   CF  9F 0014C          PUSHAB  DO_DELETE
                           B0    AD  9F 00150          PUSHAB  FAB
                 00000000G   00    0C  FB 00153          CALLS   #12, LIB$FILE_SCAN
                      83     AD    80  8F 8A 0015A      BICB2   #128, NAM+51                                    ; 0560
                    FF58     CD        08  88 0015F     BISB2   #8, NAM+8                                       ; 0561
                         FF52  CD        B4 00164       CLRW    NAM+2                                           ; 0565
                         FF5A  CD        B4 00168       CLRW    NAM+10                                          ; 0564
                         FF60  CD        D4 0016C       CLRL    NAM+16                                          ; 0566
                           E4  AD        B4 00170       CLRW    FAB+52                                          ; 0567
                           B0  AD        9F 00173       PUSHAB  FAB                                             ; 0569
                 00000000G   00        01  FB 00176    CALLS   #1, SYS$PARSE
                             04    AE  D5 0017D         TSTL    WORST_ERROR                                     ; 0575
                                 05  13 00180          BEQL    21$
                           50    04  AE  D0 00182      MOVL    WORST_ERROR, RET_STATUS                         ; 0577
                                 04  00186             RET
                           50        01  D0 00187 21$:  MOVL    #1, RET_STATUS                                  ; 0579
                                 04  0018A             RET                                                     ; 0583
                                0000 0018B 22$:     .WORD   Save nothing                                        ; 0401
                           50    08  AC  D0 0018D      MOVL    8(AP), R0
                           50    04  A0  D0 00191      MOVL    4(R0), R0
                         FCEC    C0  9F 00195          PUSHAB  INTERCEPT_FLAG
                                 01  DD 00199          PUSHL   #1
                                 5E  DD 0019B          PUSHL   SP
                       7E          04  AC  7D 0019D     MOVQ    4(AP), -(SP)
                0000V  CF              03  FB 001A1     CALLS   #3, DELETE_HANDLER
                                       04  001A6       RET

; Routine Size:  423 bytes,    Routine Base:  _LIB$CODE + 0000
```

```
;  517          0584  1 %SBTTL 'DO_DELETE - Delete a file'
;  518          0585  1 ROUTINE DO_DELETE (
;  519          0586  1     FAB: REF $FAB_DECL,                        ! Next input file FAB
;  520          0587  1     UNUSED_1,                                  ! Unused here
;  521          0588  1     UNUSED_2,                                  ! Unused here
;  522          0589  1     UNUSED_3,                                  ! Unused here
;  523          0590  1     SUCCESS_ROUTINE,                           ! Success routine address
;  524          0591  1     UNUSED_4,                                  ! Unused here
;  525          0592  1     CONFIRM_ROUTINE,                           ! Confirm routine address
;  526          0593  1     USER_ARG,                                  ! User argument
;  527          0594  1     RESULTANT_NAME,                            ! Resultant filename
;  528          0595  1     UNUSED_5,                                  ! Unused here
;  529          0596  1     ERROR_SOURCE: REF VECTOR [, LONG],         ! Source of error
;  530          0597  1     INTERCEPT_FLAG: REF VECTOR [, LONG]        ! Intercept flag
;  531          0598  1     ): NOVALUE =
;  532          0599  1
;  533          0600  1 !++
;  534          0601  1 ! FUNCTIONAL DESCRIPTION:
;  535          0602  1 !
;  536          0603  1 !       This routine is called once for each file that LIB$FILE_SCAN
;  537          0604  1 !       finds.  It deletes the file described by FAB.
;  538          0605  1 !
;  539          0606  1 ! CALLING SEQUENCE:
;  540          0607  1 !
;  541          0608  1 !       DO_DELETE is called with the same arguments as was LIB$FILE_SCAN
;  542          0609  1 !       in LIB$DELETE_FILE.
;  543          0610  1 !
;  544          0611  1 ! FORMAL PARAMETERS:
;  545          0612  1 !
;  546          0613  1 !       See LIB$DELETE_FILE for a description of all parameters.
;  547          0614  1 !
;  548          0615  1 ! IMPLICIT INPUTS:
;  549          0616  1 !
;  550          0617  1 !       NONE
;  551          0618  1 !
;  552          0619  1 ! IMPLICIT OUTPUTS:
;  553          0620  1 !
;  554          0621  1 !       NONE
;  555          0622  1 !
;  556          0623  1 ! COMPLETION STATUS:
;  557          0624  1 !
;  558          0625  1 !       NONE
;  559          0626  1 !
;  560          0627  1 ! SIDE EFFECTS:
;  561          0628  1 !
;  562          0629  1 !       Deletes a file.
;  563          0630  1 !
;  564          0631  1 !--
;  565          0632  1
;  566          0633  2     BEGIN
;  567          0634  2
;  568          0635  2     LOCAL
;  569          0636  2         NAM: REF $NAM_DECL,                    ! NAM of FAB
;  570          0637  2         LOCAL_DSC: BLOCK [8, BYTE],            ! Local string descriptor
;  571          0638  2         STRING_PTR,                            ! Pointer to string used
;  572          0639  2         COPY_STATUS;                           ! Status from LIB$SCOPY
;  573          0640  2
```

```
;  574          0641  2     BUILTIN
;  575          0642  2         AP,
;  576          0643  2         CALLG;
;  577          0644  2
;  578          0645  2     !+
;  579          0646  2     ! Set NAM address.
;  580          0647  2     !-
;  581          0648  2
;  582          0649  2     NAM = .FAB [FAB$L_NAM];
;  583          0650  2
;  584          0651  2     !+
;  585          0652  2     ! Copy the resultant file name to the user's string, if specified.
;  586          0653  2     ! If we get an error from the copy, signal it.
;  587          0654  2     ! If no user string, use our own.
;  588          0655  2     !-
;  589          0656  2
;  590          0657  2     STRING_PTR = .RESULTANT_NAME;
;  591          0658  2     IF .STRING_PTR NEQA 0
;  592          0659  2     THEN
;  593          0660  3         BEGIN
;  594          0661  3         COPY_STATUS = LIB$SCOPY_R_DX (%REF(.NAM [NAM$B_RSL]),
;  595          0662  3             .NAM [NAM$L_RSA], .RESULTANT_NAME);
;  596          0663  3         IF NOT .COPY_STATUS
;  597          0664  3         THEN
;  598          0665  4             BEGIN
;  599          0666  4             INTERCEPT_FLAG [0] = 1;
;  600          0667  4             SIGNAL_STOP (.COPY_STATUS);
;  601          0668  4             RETURN;
;  602          0669  3             END;
;  603          0670  3         END
;  604          0671  2     ELSE
;  605          0672  3         BEGIN
;  606          0673  3         STRING_PTR = LOCAL_DSC;          ! Use our string
;  607          0674  3         LOCAL_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  608          0675  3         LOCAL_DSC [DSC$B_CLASS] = DSC$K_CLASS_S;
;  609          0676  3         LOCAL_DSC [DSC$W_LENGTH] = .NAM [NAM$B_RSL];
;  610          0677  3         LOCAL_DSC [DSC$A_POINTER] = .NAM [NAM$L_RSA];
;  611          0678  2         END;
;  612          0679  2
;  613          0680  2     !+
;  614          0681  2     ! If there is a CONFIRM_ROUTINE, ask it if it wants the file deleted.
;  615          0682  2     ! Note use of linkage name BLISS to perform a general-routine-call.
;  616          0683  2     !-
;  617          0684  2
;  618          0685  2     IF .CONFIRM_ROUTINE NEQA 0
;  619          0686  2     THEN
;  620          0687  2         IF NOT BLISS (.CONFIRM_ROUTINE, .STRING_PTR, .FAB, .USER_ARG)
;  621          0688  2         THEN
;  622          0689  2             RETURN;
;  623          0690  2
;  624          0691  2     !+
;  625          0692  2     ! Do the $ERASE.
;  626          0693  2     !-
;  627          0694  2
;  628          0695  3     IF NOT $ERASE (FAB = FAB [0,0,0,0])
;  629          0696  2     THEN
;  630          0697  3         BEGIN
```

```
0698 3          !+
0699 3          ! Call the DELETE_ERROR routine with the same arguments we were
0700 3          ! called with.  It will call the user's error routine, if any.
0701 3          ! Set ERROR_SOURCE to 1 so that DELETE_ERROR will know the error
0702 3          ! came from the $ERASE.
0703 3          !-
0704 3          ERROR_SOURCE [0] = 1;
0705 3          CALLG (.AP, DELETE_ERROR);
0706 3          RETURN;
0707 2          END;
0708 2
0709 2      !+
0710 2      ! If there is a success routine, call it.
0711 2      !-
0712 2
0713 2      IF .SUCCESS_ROUTINE NEQA 0
0714 2      THEN
0715 2          BLISS (.SUCCESS_ROUTINE, .STRING_PTR, .USER_ARG);
0716 2
0717 2      RETURN;
0718 2
0719 1      END;                                    ! End of routine DO_DELETE


                                                .EXTRN  SYS$ERASE

                                001C 00000 DO_DELETE:
                                                .WORD   Save R2,R3,R4                   ; 0585
         5E           0C  C2 00002             SUBL2   #12, SP                         ;
         53      04   AC  D0 00005             MOVL    FAB, R3                         ; 0649
         52      28   A3  D0 00009             MOVL    40(R3), NAM                     ;
         54      24   AC  D0 0000D             MOVL    RESULTANT_NAME, STRING_PTR      ; 0657
                      26  13 00011             BEQL    1$                              ; 0658
                 24   AC  DD 00013             PUSHL   RESULTANT_NAME                  ; 0662
                 04   A2  DD 00016             PUSHL   4(NAM)                          ;
      08 AE      03   A2  9A 00019             MOVZBL  3(NAM), 8(SP)                   ; 0661
                 08   AE  9F 0001E             PUSHAB  8(SP)                           ;
00000000G 00          03  FB 00021             CALLS   #3, LIB$SCOPY_R_DX              ;
         22           50  E8 00028             BLBS    COPY_STATUS, 2$                 ; 0663
      30 BC           01  D0 0002B             MOVL    #1, @INTERCEPT_FLAG             ; 0666
                      50  DD 0002F             PUSHL   COPY_STATUS                     ; 0667
00000000G 00          01  FB 00031             CALLS   #1, LIB$STOP                    ;
                          04 00038             RET                                     ; 0665
         54      04   AE  9E 00039 1$:         MOVAB   LOCAL_DSC, STRING_PTR           ; 0673
      06 AE    010E   8F  B0 0003D             MOVW    #270, LOCAL_DSC+2               ; 0674
      04 AE      03   A2  9B 00043             MOVZBW  3(NAM), LOCAL_DSC               ; 0676
      08 AE      04   A2  D0 00048             MOVL    4(NAM), LOCAL_DSC+4             ; 0677
                 1C   AC  D5 0004D 2$:         TSTL    CONFIRM_ROUTINE                 ; 0685
                      0E  13 00050             BEQL    3$                              ;
                 20   AC  DD 00052             PUSHL   USER_ARG                        ; 0687
                      53  DD 00055             PUSHL   R3                              ;
                      54  DD 00057             PUSHL   STRING_PTR                      ;
      1C BC           03  FB 00059             CALLS   #3, @CONFIRM_ROUTINE            ;
         24           50  E9 0005D             BLBC    R0, 5$                          ;
                      53  DD 00060 3$:         PUSHL   R3                              ; 0695
00000000G 00          01  FB 00062             CALLS   #1, SYS$ERASE                   ;
```

```
                   0A          50  E8 00069      BLBS    R0, 4$                          ;
            2C     BC          01  D0 0006C      MOVL    #1, @ERROR_SOURCE               ; 0704
         0000V     CF          6C  FA 00070      CALLG   (AP), DELETE_ERROR              ; 0705
                                   04 00075      RET                                     ; 0697
                         14    AC  D5 00076 4$:  TSTL    SUCCESS_ROUTINE                 ; 0713
                               09  13 00079      BEQL    5$                              ;
                         20    AC  DD 0007B      PUSHL   USER_ARG                        ; 0715
                               54  DD 0007E      PUSHL   STRING_PTR                      ;
            14     BC          02  FB 00080      CALLS   #2, @SUCCESS_ROUTINE            ;
                                   04 00084 5$:  RET                                     ; 0719

; Routine Size:  133 bytes,    Routine Base:  _LIB$CODE + 01A7
```

```
;  654     0720  1 %SBTTL 'DELETE_ERROR - Report error during delete'
;  655     0721  1 ROUTINE DELETE_ERROR (
;  656     0722  1     FAB: REF $FAB_DECL,                     ! Next input file FAB
;  657     0723  1     UNUSED_1,                               ! Unused here
;  658     0724  1     UNUSED_2,                               ! Unused here
;  659     0725  1     UNUSED_3,                               ! Unused here
;  660     0726  1     UNUSED_4,                               ! Unused here
;  661     0727  1     ERROR_ROUTINE,                          ! User's error routine
;  662     0728  1     UNUSED_5,                               ! Unused here
;  663     0729  1     USER_ARG,                               ! User's argument
;  664     0730  1     RESULTANT_NAME,                         ! Resultant file name
;  665     0731  1     WORST_ERROR: REF BLOCK [, BYTE],        ! Worst previous error
;  666     0732  1     ERROR_SOURCE: REF VECTOR [, LONG],      ! Code for source of error
;  667     0733  1     INTERCEPT_FLAG: REF VECTOR [, LONG]     ! Signal intercept flag
;  668     0734  1     ): NOVALUE =
;  669     0735  1
;  670     0736  1 !++
;  671     0737  1 ! FUNCTIONAL DESCRIPTION:
;  672     0738  1 !
;  673     0739  1 !       This routine is called when LIB$FILE_SCAN detects an error.
;  674     0740  1 !       It calls the user's error routine, if one exists.
;  675     0741  1 !       If the user's error routine returns success, or if there is
;  676     0742  1 !       no user error routine, processing of remaining files continues.
;  677     0743  1 !       If the user's error routine returns failure, the error is
;  678     0744  1 !       signalled.  This signal is converted to the return status of
;  679     0745  1 !       LIB$DELETE_FILE.
;  680     0746  1 !
;  681     0747  1 ! CALLING SEQUENCE:
;  682     0748  1 !
;  683     0749  1 !       Called from LIB$FILE_SCAN and DO_DELETE with the same arguments
;  684     0750  1 !       as passed to LIB$FILE_SCAN from LIB$DELETE_FILE.
;  685     0751  1 !
;  686     0752  1 ! FORMAL PARAMETERS:
;  687     0753  1 !
;  688     0754  1 !       See body of LIB$DELETE_FILE for descriptions of arguments.
;  689     0755  1 !
;  690     0756  1 ! IMPLICIT INPUTS:
;  691     0757  1 !
;  692     0758  1 !       NONE
;  693     0759  1 !
;  694     0760  1 ! IMPLICIT OUTPUTS:
;  695     0761  1 !
;  696     0762  1 !       NONE
;  697     0763  1 !
;  698     0764  1 ! COMPLETION STATUS:
;  699     0765  1 !
;  700     0766  1 !       NONE
;  701     0767  1 !
;  702     0768  1 ! SIDE EFFECTS:
;  703     0769  1 !
;  704     0770  1 !       Signals .FAB [FAB$L_STS]
;  705     0771  1 !
;  706     0772  1 !--
;  707     0773  1
;  708     0774  2     BEGIN
;  709     0775  2
;  710     0776  2     LOCAL
```

```
;  711     0777  2          NAM: REF $NAM_DECL,                    ! NAM block
;  712     0778  2          LOCAL_DSC: BLOCK [8, BYTE],            ! Our filename descriptor
;  713     0779  2          STRING_PTR,                            ! Pointer to user's string or our own
;  714     0780  2          COPY_STATUS;                           ! Status from LIB$SCOPY
;  715     0781  2
;  716     0782  2      !+
;  717     0783  2      ! Build our local descriptor to contain the filespec    the file in error.
;  718     0784  2      ! Note that the filespec is guaranteed to be in the h .N because
;  719     0785  2      ! LIB$FILE_SCAN puts it there before calling us.
;  720     0786  2      !-
;  721     0787  2
;  722     0788  2      NAM = .FAB [FAB$L_NAM];
;  723     0789  2      LOCAL_DSC [DSC$B_DTYPE] = DSC$K_DTYPE_T;
;  724     0790  2      LOCAL_DSC [DSC$B_CLASS] = DSC$K_CLASS_S;
;  725     0791  2      LOCAL_DSC [DSC$W_LENGTH] = .NAM [NAM$B_RSL];
;  726     0792  2      LOCAL_DSC [DSC$A_POINTER] = .NAM [NAM$[_RSA];
;  727     0793  2
;  728     0794  2      !+
;  729     0795  2      ! If the user has specified DELETED_FILE, copy the filename to it.
;  730     0796  2      !-
;  731     0797  2
;  732     0798  2      STRING_PTR = .RESULTANT_NAME;
;  733     0799  2      IF .STRING_PTR NEQA 0
;  734     0800  2      THEN
;  735     0801  3          BEGIN
;  736     0802  3          COPY_STATUS = LIB$SCOPY_DXDX (LOCAL_DSC, .RESULTANT_NAME);
;  737     0803  3          IF NOT .COPY_STATUS
;  738     0804  3          THEN
;  739     0805  4              BEGIN
;  740     0806  4              INTER.EPT_FLAG [0] = 1;      ! Cause handler to intercept signal
;  741     0807  4              SIGNAL_STOP (.COPY_STATUS);
;  742     0808  4              RETURN;
;  743     0809  3              END;
;  744     0810  3          END
;  745     0811  2      ELSE
;  746     0812  2          STRING_PTR = LOCAL_DSC;          ! Use our own string
;  747     0813  2
;  748     0814  2      !+
;  749     0815  2      ! If a user error routine has been specified, call it with arguments
;  750     0816  2      ! of the filename, STS, STV, error source and user argument.  If it returns
;  751     0817  2      ! failure, signal the error.  Since the user's routine was called,
;  752     0818  2      ! store LIB$_ERRROUCAL in WORST_ERROR.
;  753     0819  2      !-
;  754     0820  2
;  755     0821  2      IF .ERROR_ROUTINE NEQA 0
;  756     0822  2      THEN
;  757     0823  3          BEGIN
;  758     0824  3          IF NOT BLISS (.ERROR_ROUTINE, .STRING_PTR,
;  759     0825  3              FAB [FAB$L_STS], FAB [FAB$L_STV], ERROR_SOURCE [0], .USER_ARG)
;  760     0826  3          THEN
;  761     0827  4              BEGIN
;  762     0828  4              INTERCEPT_FLAG [0] = 1;      ! Cause handler to intercept signal
;  763     0829  4              SIGNAL_STOP (.FAB [FAB$L_STS]);
;  764     0830  4              RETURN;
;  765     0831  3              END;
;  766     0832  3          WORST_ERROR [0,0,32,0] = LIB$_ERRROUCAL;        ! Error routine called
;  767     0833  3          END
```

```
2       ELSE
3           BEGIN
3           !+
3           ! If this error is worse than any previous errors, store it in
3           ! WORST_ERROR.  (Use GEQU so that the initial zero gets replaced if
3           ! the error is a warning.)
3           !-
3
3           IF .BLOCK [FAB [FAB$L_STS], STS$V_SEVERITY;4, BYTE] GEQU
3               .WORST_ERROR [STS$V_SEVERITY]
3           THEN
3               WORST_ERROR [0,0,32,0] = .FAB [FAB$L_STS];
2           END;
2
2       !+
2       ! Reset ERROR_SOURCE to zero so that future $SEARCH errors will be
2       ! properly indicated.
2       !-
2
2       ERROR_SOURCE [0] = 0;
2
2       RETURN;
2
1       END;                                    ! End of routine DELETE_ERROR
```

```
                                 000C 00000 DELETE_ERROR:
                                                 .WORD   Save R2,R3                          ; 0721
            5E           08  C2 00002            SUBL2   #8, SP                              ;
            52      04   AC  D0 00005            MOV_    FAB, R2                             ; 0788
            50      28   A2  D0 00009            MOVL    40(R2), NAM                         ;
        02  AE    010E   8F  B0 0000D            MOVW    #270, LOCAL_DSC+2                   ; 0789
            6E      03   A0  9B 00013            MOVZBW  3(NAM), LOCAL_DSC                   ; 0791
        04  AE      04   A0  D0 00017            MOVL    4(NAM), LOCAL_DSC+4                 ; 0792
            53      24   AC  D0 0001C            MOVL    RESULTANT_NAME, STRING_PTR          ; 0798
                         18  13 00020            BEQL    1$                                  ; 0799
                    24   AC  DD 00022            PUSHL   RESULTANT_NAME                      ; 0802
                    04   AE  9F 00025            PUSHAB  LOCAL_DSC                           ;
 00000000G  00           02  FB 00028            CALLS   #2, LIB$SCOPY_DXDX                  ;
            0B           50  E8 0002F            BLBS    COPY_STATUS, 2$                     ; 0803
        30  BC           01  D0 00032            MOVL    #1, @INTERCEPT_FLAG                 ; 0806
                         50  DD 00036            PUSHL   COPY_STATUS                         ; 0807
                         24  11 00038            BRB     3$                                  ;
            53           6E  9E 0003A 1$:        MOVAB   LOCAL_DSC, STRING_PTR               ; 0812
                    18   AC  D5 0003D 2$:        TSTL    ERROR_ROUTINE                       ; 0821
                         2E  13 00040            BEQL    5$                                  ;
                    20   AC  DD 00042            PUSHL   USER_ARG                            ; 0825
                    2C   AC  DD 00045            PUSHL   ERROR_SOURCE                        ;
                    0C   A2  9F 00048            PUSHAB  12(R2)                              ;
                    08   A2  9F 0004B            PUSHAB  8(R2)                               ;
                         53  DD 0004E            PUSHL   STRING_PTR                          ;
        18  BC           05  FB 00050            CALLS   #5, @ERROR_ROUTINE                  ;
            0F           50  E8 00054            BLBS    R0, 4$                              ;
        30  BC           01  D0 00057            MOVL    #1, @INTERCEPT_FLAG                 ; 0828
```

```
                                 08     A2  DD 0005B       PUSHL   8(R2)                                 ; 0829
                 00000000G  00          01  FB 0005E 3$:   CALLS   #1, LIB$STOP                          ;
                                            04 00065       RET                                           ; 0827
                       28   BC 00000000G 8F  D0 00066 4$:   MOVL    #LIB$_ERRROUCAL, @WORST_ERROR         ; 0832
                                        13  11 0006E       BRB     6$                                    ; 0821
     50       28   BC          03       00  EF 00070 5$:   EXTZV   #0, #3, @WORST_ERROR, R0              ; 0843
     50       08   A2          03       00  ED 00076       CMPZV   #0, #3, 8(R2), R0                     ;
                                        05  1F 0007C       BLSSU   6$                                    ;
                       28   BC   08     A2  D0 0007E       MOVL    8(R2), @WORST_ERROR                   ; 0845
                                 2C     BC  D4 00083 6$:   CLRL    @ERROR_SOURCE                         ; 0853
                                            04 00086       RET                                           ; 0857

; Routine Size:  135 bytes,    Routine Base:  _LIB$CODE + 022C
```

```
;   793          0858  1 %SBTTL 'DELETE_HANDLER - Local condition handler'
;   794          0859  1 ROUTINE DELETE_HANDLER (
;   795          0860  1         SIGNAL_ARGS: REF BLOCK [, BYTE],        ! Signal arguments array
;   796          0861  1         MECH_ARGS: REF BLOCK [, BYTE],          ! Mechanism arguments array
;   797          0862  1         ENABLE_ARGS: REF VECTOR [, LONG]        ! Enable arguments array
;   798          0863  1     ) =
;   799          0864  1
;   800          0865  1 !++
;   801          0866  1 ! FUNCTIONAL DESCRIPTION:
;   802          0867  1 !
;   803          0868  1 !       This is the condition handler enabled by LIB$DELETE_FILE.
;   804          0869  1 !       If this is not an unwind, and if the INTERCEPT_FLAG enable
;   805          0870  1 !       argument is set, then LIB$SIG_TO_RET is called to convert the
;   806          0871  1 !       signal to a return status.
;   807          0872  1 !
;   808          0873  1 ! CALLING SEQUENCE:
;   809          0874  1 !
;   810          0875  1 !       status.wlc.v = DELETE_HANDLER (SIGNAL_ARGS.rl.ra, MECH_ARGS.rl.ra
;   811          0876  1 !                                      , ENABLE_ARGS.rl.ra)
;   812          0877  1 !
;   813          0878  1 ! FORMAL PARAMETERS:
;   814          0879  1 !
;   815          0880  1 !       SIGNAL_ARGS     - The signal argument list.
;   816          0881  1 !
;   817          0882  1 !       MECH_ARGS       - The mechanism argument list.
;   818          0883  1 !
;   819          0884  1 !       ENABLE_ARGS     - The enable argument list.  The one enable
;   820          0885  1 !                         argument is the address of INTERCEPT_FLAG;
;   821          0886  1 !
;   822          0887  1 ! IMPLICIT INPUTS:
;   823          0888  1 !
;   824          0889  1 !       NONE
;   825          0890  1 !
;   826          0891  1 ! IMPLICIT OUTPUTS:
;   827          0892  1 !
;   828          0893  1 !       NONE
;   829          0894  1 !
;   830          0895  1 ! ROUTINE VALUE:
;   831          0896  1 !
;   832          0897  1 !       SS$_RESIGNAL
;   833          0898  1 !
;   834          0899  1 ! SIDE EFFECTS:
;   835          0900  1 !
;   836          0901  1 !       May cause an unwind.
;   837          0902  1 !
;   838          0903  1 !--
;   839          0904  1
;   840          0905  2     BEGIN
;   841          0906  2
;   842          0907  2     BUILTIN
;   843          0908  2         AP,     ! Argument pointer
;   844          0909  2         CALLG;  ! CALLG instruction
;   845          0910  2
;   846          0911  2     !+
;   847          0912  2     ! Determine if this is an unwind.  If not, then if INTERCEPT_FLAG
;   848          0913  2     ! is set, turn this signal into an unwind.
;   849          0914  2     !-
```

```
;  850        0915  2
;  851        0916  2      IF .SIGNAL_ARGS [CHF$L_SIG_NAME] NEQU SS$_UNWIND
;  852        0917  2      THEN
;  853        0918  2          IF ..ENABLE_ARGS [1]    ! Is INTERCEPT_FLAG set?
;  854        0919  2          THEN
;  855        0920  2              CALLG (.AP, LIB$SIG_TO_RET);        ! Convert signal to return status
;  856        0921  2
;  857        0922  2      RETURN SS$_RESIGNAL;                        ! Resignal error
;  858        0923  2
;  859        0924  1      END;                                        ! End of routine DELETE_HANDLER


                                          0000 00000 DELETE_HANDLER:
                                                          .WORD   Save nothing                          ; 0859
                          50        04   AC  D0 00002     MOVL    SIGNAL_ARGS, R0                       ; 0916
                 00000920 8F        04   A0  D1 00006     CMPL    4(R0), #2336                          ;
                                         0F  13 0000E     BEQL    1$                                    ;
                          50        0C   AC  D0 00010     MOVL    ENABLE_ARGS, R0                       ; 0918
                          07        04   B0  E9 00014     BLBC    @4(R0), 1$                            ;
                 00000000G 00            6C  FA 00018     CALLG   (AP), LIB$SIG_TO_RET                  ; 0920
                          50      0918   8F  3C 0001F 1$: MOVZWL  #2328, R0                             ; 0922
                                             04 00024     RET                                           ; 0924

; Routine Size:  37 bytes,    Routine Base:  _LIB$CODE + 02B3


;  860        0925  1
```

```
;  862          0926  1 END                                        ! End of module LIB$DELETE_FILE
;  863          0927  1
;  864          0928  0 ELUDOM


                                                    .EXTRN  LIB$STOP

;                           PSECT SUMMARY
;
;      Name                       Bytes                    Attributes
;
;  _LIB$CODE                         728  NOVEC,NOWRT,  RD ,  EXE,  SHR,  LCL,  REL,  CON,  PIC,ALIGN(2)


;                       Library Statistics
;
;                                          -------- Symbols --------      Pages       Processing
;      File                                Total    Loaded   Percent      Mapped      Time
;
;  _$255$DUA28:[SYSLIB]STARLET.L32;1        9776        74         0        581        00:00.7
;  _$255$DUA28:[LIBRTL.OBJ]RTLLIB.L32;1       36         2         5          8        00:00.1


;                            COMMAND QUALIFIERS

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

; Size:          728 code + 0 data bytes
; Run Time:         00:12.4
; Elapsed Time:     00:52.2
; Lines/CPU Min:     4486
; Lexemes/CPU-Min: 33065
; Memory Used:  172 pages
; Compilation Complete
```

0205 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY