File created: 23-May-2024 23:20:49 **{DSK}<home>frank>il>medley>sources>NSFILING.;2**

edit by: frank

changes to: (FNS \NSRANDOM.CREATE.STREAM \NSFILING.GETFILE)

previous date: 28-Jun-99 17:07:34 {DSK}<home>frank>il>medley>sources>NSFILING.;1

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

```
(RPAQQ NSFILINGCOMS
       [(COMS                                                    ; Filing Protocol
              (COURIERPROGRAMS FILING FILING.4)
              (DECLARE%: EVAL@COMPILE DONTCOPY (CONSTANTS * NSFILINGCONSTANTS)
                     (RECORDS NSFILINGSTREAM FILINGSESSION FILINGHANDLE NSFILESERVER NSFILINGDEVICEINFO
                            \NSFILING.GENFILESTATE NSFILINGPARSE NSPAGECACHE)
                     (MACROS WITHOUT.SESSION.MONITOR)
                     (GLOBALVARS \NSFILING.CONNECTIONS \NSFILING.DEVICE \NSFILING.NULL.HANDLE
                            \NSFILING.ATTRIBUTES \LISP.TO.NSFILING.ATTRIBUTES \NSFILING.USEFUL.ATTRIBUTE.TYPES
                            \NSFILING.PROGRAM.NAME \NSFILING.ACTIVE.SESSIONS FILING.CACHE.LIMIT
                            *NSFILING-PAGE-CACHE-INCREMENT* *NSFILING-PAGE-CACHE-LIMIT* *NSFILING-RANDOM-ACCESS*
                            *NSFILING-SESSION-TIMEOUT* \NSRANDOM.CHECK.CACHE \NSFILING.PROTECTION.BITS
                            \FILEDEVICES)
                     (LOCALVARS . T)
                     (FILES (SOURCE)
                            SPPDECLS)
                     (FILES (LOADCOMP)
                            COURIER))
              (INITRECORDS FILINGSESSION FILINGHANDLE)
              (FNS \FILINGSESSION.DEFPRINT \FILINGHANDLE.DEFPRINT))
        [COMS (FNS \GET.FILING.ATTRIBUTE \PUT.FILING.ATTRIBUTE \GET.SESSION.HANDLE \PUT.SESSION.HANDLE)
              (PROP COURIERDEF FILING.SESSION FILING.ATTRIBUTE)
              (DECLARE%: EVAL@COMPILE DOCOPY (VARS \NSFILING.NULL.HANDLE \NSFILING.PROTECTION.BITS
                                                   \NSFILING.ATTRIBUTES \LISP.TO.NSFILING.ATTRIBUTES
                                                   (\NSFILING.USEFUL.ATTRIBUTE.TYPES (
                                                                                      \FILING.ATTRIBUTE.TYPE.SEQUENCE
                                                                                          '(CREATED.ON FILE.ID
                                                                                                  IS.DIRECTORY
                                                                                                  PATHNAME
                                                                                                  SIZE.IN.BYTES
                                                                                                  FILE.TYPE VERSION]
        (INITVARS (FILING.CACHE.LIMIT 6)
               (NSFILING.SHOW.STATUS T)
               (FILING.ENUMERATION.DEPTH T)
               (\NSFILING.LOCK (CREATE.MONITORLOCK 'NSFILING))
               (\NSFILING.PROGRAM.NAME 'FILING)
               (\NSFILING.ACTIVE.SESSIONS)
               (*NSFILING-RANDOM-ACCESS* T)
               (*NSFILING-PAGE-CACHE-LIMIT* 8)
               (*NSFILING-PAGE-CACHE-INCREMENT* 4)
               (*NSFILING-SESSION-TIMEOUT* '(900 . 21600))
               (\NSRANDOM.CHECK.CACHE))
        (COMS                                                    ; Connection maintenance
              (FNS \GETFILINGCONNECTION \NSFILING.GET.NEW.SESSION \NSFILING.GET.STREAM \NSFILING.COURIER.OPEN
                   \NSFILING.CLOSE.BULKSTREAM \NSFILING.RELEASE.BULKSTREAM FILING.CALL \NSFILING.LOGIN
                   \NSFILING.AFTER.LOGIN \NSFILING.SET.CONTINUANCE \NSFILING.LOGOUT \NSFILING.DISCARD.SESSION
                   \VALID.FILING.CONNECTIONP \NSFILING.CLOSE.CONNECTIONS BREAK.NSFILING.CONNECTION)
              (ADDVARS (\AFTERLOGINFNS \NSFILING.AFTER.LOGIN)))
        (COMS                                                    ; Support
              (FNS \NSFILING.CONNECT \NSFILING.MAYBE.CREATE \NSFILING.REMOVEQUOTES \NSFILING.ADDQUOTES
                   \FILING.ATTRIBUTE.TYPE.SEQUENCE \FILING.ATTRIBUTE.TYPE \LISP.TO.NSFILING.ATTRIBUTE))
        (COMS                                                    ; FILINGHANDLE stuff
              (FNS \NSFILING.GETFILE \NSFILING.LOOKUP.CACHE \NSFILING.ADD.TO.CACHE \NSFILING.OPEN.HANDLE
                   \NSFILING.CONFLICTP \NSFILING.CHECK.ACCESS \NSFILING.FILLIN.ATTRIBUTES
                   \NSFILING.COMPOSE.PATHNAME \NSFILING.PARSE.FILENAME \NSFILING.ERRORHANDLER
                   \NSFILING.WHENCLOSED \NSFILING.CLOSE.HANDLE \NSFILING.FULLNAME))
        (COMS                                                    ; NSFILING device
              (FNS \NSFILING.OPENFILE \NSFILING.HANDLE.ERROR \NSFILING.CLOSEFILE \NSFILING.EVENTFN
                   \NSFILING.DELETEFILE \NSFILING.CHILDLESS-P \NSFILING.DIRECTORYNAMEP \NSFILING.HOSTNAMEP
                   \NSFILING.GETFILENAME \NSFILING.GETFILEINFO \NSFILING.GET.ATTRIBUTES
                   \NSFILING.GETFILEINFO.FROM.PLIST \NSFILING.GDATE \NSFILING.SETFILEINFO \NSFILING.GET/SETINFO
                   \NSFILING.UPDATE.ATTRIBUTES \NSFILING.GETEOFPTR \NSFILING.GENERATEFILES
                   \NSFILING.GENERATE.STARS \NSFILING.NEXTFILE \NSFILING.FILEINFOFN \NSFILING.RENAMEFILE
                   \NSFILING.COPYFILE \NSFILING.COPY/RENAME))
        (COMS                                                    ; Random access methods
              (FNS \NSRANDOM.CLOSEFILE \NSRANDOM.RELEASE.HANDLE \NSRANDOM.RELEASE.LOCK
                   \NSRANDOM.RELEASE.IF.ERROR \NSRANDOM.CREATE.STREAM \NSRANDOM.READPAGES \NSRANDOM.READ.SEGMENT
                   \NSRANDOM.PREPARE.CACHE \NSRANDOM.FETCH.CACHE \NSRANDOM.CHECK.CACHE \NSRANDOM.WRITEPAGES
                   \NSRANDOM.WRITE.SEGMENT \NSRANDOM.WROTE.HANDLE \NSRANDOM.SETEOFPTR \NSRANDOM.TRUNCATEFILE
                   \NSRANDOM.UPDATE.VALIDATION \NSRANDOM.OPENFILE)
                                                                 ; error handling
              (FNS \NSRANDOM.HANDLE.ERROR \NSRANDOM.PROCEEDABLE.ERROR \NSRANDOM.REESTABLISH
                   \NSRANDOM.STREAM.CHANGED \NSRANDOM.DESTROY.STREAM \NSRANDOM.SESSION.WATCHER
                   \NSRANDOM.ENSURE.WATCHER))
```

```
        (COMS                                                          ; Cleaning up directories
              (FNS GC-FILING-DIRECTORY \NSGC.COLLECT.DIRECTORIES))
        (COMS                                                          ; Deserialize (special for NSMAIL)
              (FNS \NSFILING.DESERIALIZE \NSFILING.DESERIALIZE1))
        [COMS (FNS \NSFILING.INIT)
              (DECLARE%: DONTEVAL@LOAD DOCOPY (P (\NSFILING.INIT]
        (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
                                                                              (NLAML)
                                                                              (LAMA FILING.CALL])
```

;; Filing Protocol

```
(COURIERPROGRAM FILING (10 5)
    TYPES
      [(ATTRIBUTE.TYPE LONGCARDINAL)
       (ATTRIBUTE.TYPE.SEQUENCE (SEQUENCE ATTRIBUTE.TYPE))
       (ATTRIBUTE FILING.ATTRIBUTE)
       (ATTRIBUTE.SEQUENCE (SEQUENCE FILING.ATTRIBUTE))
       (CONTROL.TYPE (ENUMERATION (LOCK 0)
                                  (TIMEOUT 1)
                                  (ACCESS 2)))
       (CONTROL.TYPE.SEQUENCE (SEQUENCE CONTROL.TYPE))
       (CONTROL (CHOICE (LOCK 0 LOCK)
                        (TIMEOUT 1 TIMEOUT)
                        (ACCESS 2 ACCESS.SEQUENCE)))
       (CONTROL.SEQUENCE (SEQUENCE CONTROL))
       (LOCK (ENUMERATION (NONE 0)
                          (SHARE 1)
                          (EXCLUSIVE 2)))
       (TIMEOUT CARDINAL)
       (ACCESS.TYPE (ENUMERATION (READ 0)
                                 (WRITE 1)
                                 (OWNER 2)
                                 (ADD 3)
                                 (REMOVE 4)
                                 (ALL 65535)))
       (ACCESS.SEQUENCE (SEQUENCE ACCESS.TYPE))
       (ACCESS.ENTRY (RECORD (KEY (CLEARINGHOUSE . NAME))
                             (ACCESS ACCESS.SEQUENCE)))
       (ACCESS.LIST (RECORD (ENTRIES (SEQUENCE ACCESS.ENTRY))
                            (DEFAULTED BOOLEAN)))
       (SCOPE.TYPE (ENUMERATION (COUNT 0)
                                (DIRECTION 1)
                                (FILTER 2)
                                (DEPTH 3)))
       (SCOPE (CHOICE (COUNT 0 CARDINAL)
                      (DIRECTION 1 DIRECTION)
                      (FILTER 2 FILTER)
                      (DEPTH 3 CARDINAL)))
       (SCOPE.SEQUENCE (SEQUENCE SCOPE))
       (DIRECTION (ENUMERATION (FORWARD 0)
                               (BACKWARD 1)))
       (FILTER (CHOICE (LT 0 FILTER.ATTRIBUTE)
                       (LE 1 FILTER.ATTRIBUTE)
                       (= 2 FILTER.ATTRIBUTE)
                       (~= 3 FILTER.ATTRIBUTE)
                       (GE 4 FILTER.ATTRIBUTE)
                       (GT 5 FILTER.ATTRIBUTE)
                       (AND 6 (SEQUENCE FILTER))
                       (OR 7 (SEQUENCE FILTER))
                       (NOT 8 FILTER)
                       (NONE 9 NIL)
                       (ALL 10 NIL)
                       (MATCHES 11 ATTRIBUTE)))
       (FILTER.ATTRIBUTE (RECORD (ATTRIBUTE FILING.ATTRIBUTE)
                                 (INTERPRETATION INTERPRETATION)))
       (INTERPRETATION (ENUMERATION (NONE 0)
                                    (BOOLEAN 1)
                                    (CARDINAL 2)
                                    (LONGCARDINAL 3)
                                    (TIME 4)
                                    (INTEGER 5)
                                    (LONGINTEGER 6)
                                    (STRING 7)))
       (BYTE.RANGE (RECORD (FIRSTBYTE LONGCARDINAL)
                           (COUNT LONGCARDINAL)))
       (CREDENTIALS (AUTHENTICATION . CREDENTIALS))
       (HANDLE (ARRAY 2 UNSPECIFIED))
       (SESSION FILING.SESSION)
       (VERIFIER (AUTHENTICATION . VERIFIER))
       (SIMPLE.VERIFIER (AUTHENTICATION . SIMPLE.VERIFIER))
       (FILE.ID (ARRAY 5 UNSPECIFIED))
       (USER (CLEARINGHOUSE . NAME))
       (ORDERING (RECORD (KEY ATTRIBUTE.TYPE)
                         (ASCENDING BOOLEAN)
                         (INTERPRETATION INTERPRETATION)))
       (ARGUMENT.PROBLEM (ENUMERATION (Illegal 0)
```

```
                              (Disallowed 1)
                              (Unreasonable 2)
                              (Unimplemented 3)
                              (Duplicated 4)
                              (Missing 5)))
      (ACCESS.PROBLEM (ENUMERATION (AccessRightsInsufficient 0)
                           (AccessRightsIndeterminate 1)
                           (FileChanged 2)
                           (FileDamaged 3)
                           (FileInUse 4)
                           (FileNotFound 5)
                           (FileOpen 6)))
      (CONNECTION.PROBLEM (ENUMERATION (NoRoute 0)
                               (NoResponse 1)
                               (TransmissionHardware 2)
                               (TransportTimeout 3)
                               (TooManyLocalConnections 4)
                               (TooManyRemoteConnections 5)
                               (MissingCourier 6)
                               (MissingProgram 7)
                               (MissingProcedure 8)
                               (ProtocolMismatch 9)
                               (ParameterInconsistency 10)
                               (InvalidMessage 11)
                               (ReturnTimedOut 12)
                               (Other 65535)))
      (HANDLE.PROBLEM (ENUMERATION (Invalid 0)
                           (NullDisallowed 1)
                           (DirectoryRequired 2)))
      (INSERTION.PROBLEM (ENUMERATION (PositionUnavailable 0)
                              (FileNotUnique 1)
                              (LoopInHierarchy 2)))
      (SERVICE.PROBLEM (ENUMERATION (CannotAuthenticate 0)
                            (ServiceFull 1)
                            (ServiceUnavailable 2)
                            (SessionInUse 3)
                            (UnknownService 4)))
      (SESSION.PROBLEM (ENUMERATION (TokenInvalid 0)))
      (SPACE.PROBLEM (ENUMERATION (AllocationExceeded 0)
                          (AttributeAreaFull 1)
                          (MediumFull 2)))
      (TRANSFER.PROBLEM (ENUMERATION (Aborted 0)
                              (ChecksumIncorrect 1)
                              (FormatIncorrect 2)
                              (NoRendezvous 3)
                              (WrongDirection 4]
```

PROCEDURES

```
  ((LOGON 0 ((CLEARINGHOUSE . NAME)
             CREDENTIALS VERIFIER)
          RETURNS
          (SESSION)
          REPORTS
          (AUTHENTICATION.ERROR SERVICE.ERROR SESSION.ERROR UNDEFINED.ERROR))
      (LOGOFF 1 (SESSION)
          RETURNS NIL REPORTS (AUTHENTICATION.ERROR SERVICE.ERROR SESSION.ERROR UNDEFINED.ERROR))
      (CONTINUE 19 (SESSION)
          RETURNS
          (CARDINAL)
          REPORTS
          (AUTHENTICATION.ERROR SESSION.ERROR UNDEFINED.ERROR))
      (OPEN 2 (ATTRIBUTE.SEQUENCE HANDLE CONTROL.SEQUENCE SESSION)
          RETURNS
          (HANDLE)
          REPORTS
          (ACCESS.ERROR ATTRIBUTE.TYPE.ERROR ATTRIBUTE.VALUE.ERROR AUTHENTICATION.ERROR CONTROL.TYPE.ERROR
                CONTROL.VALUE.ERROR HANDLE.ERROR SESSION.ERROR UNDEFINED.ERROR))
      (CLOSE 3 (HANDLE SESSION)
          RETURNS NIL REPORTS (AUTHENTICATION.ERROR HANDLE.ERROR SESSION.ERROR UNDEFINED.ERROR))
      (CREATE 4 (HANDLE ATTRIBUTE.SEQUENCE CONTROL.SEQUENCE SESSION)
          RETURNS
          (HANDLE)
          REPORTS
          (ACCESS.ERROR ATTRIBUTE.TYPE.ERROR ATTRIBUTE.VALUE.ERROR AUTHENTICATION.ERROR CONTROL.TYPE.ERROR
                 CONTROL.VALUE.ERROR HANDLE.ERROR INSERTION.ERROR SESSION.ERROR SPACE.ERROR UNDEFINED.ERROR)
          )
      (DELETE 5 (HANDLE SESSION)
          RETURNS NIL REPORTS (ACCESS.ERROR AUTHENTICATION.ERROR HANDLE.ERROR SESSION.ERROR UNDEFINED.ERROR)
          )
      (GET.CONTROLS 6 (HANDLE CONTROL.TYPE.SEQUENCE SESSION)
          RETURNS
          (CONTROL.SEQUENCE)
          REPORTS
          (ACCESS.ERROR AUTHENTICATION.ERROR CONTROL.TYPE.ERROR HANDLE.ERROR SESSION.ERROR UNDEFINED.ERROR))
      (CHANGE.CONTROLS 7 (HANDLE CONTROL.SEQUENCE SESSION)
          RETURNS NIL REPORTS (ACCESS.ERROR AUTHENTICATION.ERROR CONTROL.TYPE.ERROR CONTROL.VALUE.ERROR
                                    HANDLE.ERROR SESSION.ERROR UNDEFINED.ERROR))
      (GET.ATTRIBUTES 8 (HANDLE ATTRIBUTE.TYPE.SEQUENCE SESSION)
```

```
              RETURNS
              (ATTRIBUTE.SEQUENCE)
              REPORTS
              (ACCESS.ERROR ATTRIBUTE.TYPE.ERROR AUTHENTICATION.ERROR HANDLE.ERROR SESSION.ERROR UNDEFINED.ERROR
                     ))
       (CHANGE.ATTRIBUTES 9 (HANDLE ATTRIBUTE.SEQUENCE SESSION)
              RETURNS NIL REPORTS (ACCESS.ERROR ATTRIBUTE.TYPE.ERROR ATTRIBUTE.VALUE.ERROR AUTHENTICATION.ERROR
                                           HANDLE.ERROR INSERTION.ERROR SESSION.ERROR SPACE.ERROR UNDEFINED.ERROR)
              )
       (COPY 10 (HANDLE HANDLE ATTRIBUTE.SEQUENCE CONTROL.SEQUENCE SESSION)
             RETURNS
             (HANDLE)
             REPORTS
             (ACCESS.ERROR ATTRIBUTE.TYPE.ERROR ATTRIBUTE.VALUE.ERROR AUTHENTICATION.ERROR CONTROL.TYPE.ERROR
                    CONTROL.VALUE.ERROR HANDLE.ERROR INSERTION.ERROR SESSION.ERROR SPACE.ERROR UNDEFINED.ERROR))
       (MOVE 11 (HANDLE HANDLE ATTRIBUTE.SEQUENCE SESSION)
             RETURNS NIL REPORTS (ACCESS.ERROR ATTRIBUTE.TYPE.ERROR ATTRIBUTE.VALUE.ERROR AUTHENTICATION.ERROR
                                          HANDLE.ERROR INSERTION.ERROR SESSION.ERROR SPACE.ERROR UNDEFINED.ERROR))
       (STORE 12 (HANDLE ATTRIBUTE.SEQUENCE CONTROL.SEQUENCE BULK.DATA.SOURCE SESSION)
              RETURNS
              (HANDLE)
              REPORTS
              (ACCESS.ERROR ATTRIBUTE.TYPE.ERROR ATTRIBUTE.VALUE.ERROR AUTHENTICATION.ERROR CONNECTION.ERROR
                     CONTROL.TYPE.ERROR CONTROL.VALUE.ERROR HANDLE.ERROR INSERTION.ERROR SESSION.ERROR
                     SPACE.ERROR TRANSFER.ERROR UNDEFINED.ERROR))
       (RETRIEVE 13 (HANDLE BULK.DATA.SINK SESSION)
              RETURNS NIL REPORTS (ACCESS.ERROR AUTHENTICATION.ERROR CONNECTION.ERROR HANDLE.ERROR SESSION.ERROR
                                           TRANSFER.ERROR UNDEFINED.ERROR))
       (REPLACE 14 (HANDLE ATTRIBUTE.SEQUENCE BULK.DATA.SOURCE SESSION)
              RETURNS NIL REPORTS (ACCESS.ERROR ATTRIBUTE.TYPE.ERROR ATTRIBUTE.VALUE.ERROR AUTHENTICATION.ERROR
                                           CONNECTION.ERROR HANDLE.ERROR SESSION.ERROR SPACE.ERROR TRANSFER.ERROR
                                           UNDEFINED.ERROR))
       (SERIALIZE 15 (HANDLE BULK.DATA.SINK SESSION)
              RETURNS NIL REPORTS (ACCESS.ERROR AUTHENTICATION.ERROR CONNECTION.ERROR HANDLE.ERROR SESSION.ERROR
                                           TRANSFER.ERROR UNDEFINED.ERROR))
       (DESERIALIZE 16 (HANDLE ATTRIBUTE.SEQUENCE CONTROL.SEQUENCE BULK.DATA.SOURCE SESSION)
              RETURNS
              (HANDLE)
              REPORTS
              (ACCESS.ERROR ATTRIBUTE.TYPE.ERROR ATTRIBUTE.VALUE.ERROR AUTHENTICATION.ERROR CONNECTION.ERROR
                     CONTROL.TYPE.ERROR CONTROL.VALUE.ERROR HANDLE.ERROR INSERTION.ERROR SESSION.ERROR
                     SPACE.ERROR TRANSFER.ERROR UNDEFINED.ERROR))
       (FIND 17 (HANDLE SCOPE.SEQUENCE CONTROL.SEQUENCE SESSION)
             RETURNS
             (HANDLE)
             REPORTS
             (ACCESS.ERROR AUTHENTICATION.ERROR CONTROL.TYPE.ERROR CONTROL.VALUE.ERROR HANDLE.ERROR
                    SCOPE.TYPE.ERROR SCOPE.VALUE.ERROR SESSION.ERROR UNDEFINED.ERROR))
       (LIST 18 (HANDLE ATTRIBUTE.TYPE.SEQUENCE SCOPE.SEQUENCE BULK.DATA.SINK SESSION)
             RETURNS NIL REPORTS (ACCESS.ERROR ATTRIBUTE.TYPE.ERROR ATTRIBUTE.VALUE.ERROR AUTHENTICATION.ERROR
                                          CONNECTION.ERROR HANDLE.ERROR SCOPE.TYPE.ERROR SCOPE.VALUE.ERROR
                                          SESSION.ERROR TRANSFER.ERROR UNDEFINED.ERROR))
       (RETRIEVE.BYTES 22 (HANDLE BYTE.RANGE BULK.DATA.SINK SESSION)
              RETURNS NIL REPORTS (ACCESS.ERROR HANDLE.ERROR RANGE.ERROR SESSION.ERROR UNDEFINED.ERROR))
       (REPLACE.BYTES 23 (HANDLE BYTE.RANGE BULK.DATA.SOURCE SESSION)
              RETURNS NIL REPORTS (ACCESS.ERROR HANDLE.ERROR RANGE.ERROR SESSION.ERROR SPACE.ERROR
                                           UNDEFINED.ERROR)))
    ERRORS
      ((ATTRIBUTE.TYPE.ERROR 0 (ARGUMENT.PROBLEM ATTRIBUTE.TYPE))
       (ATTRIBUTE.VALUE.ERROR 1 (ARGUMENT.PROBLEM ATTRIBUTE.TYPE))
       (CONTROL.TYPE.ERROR 2 (ARGUMENT.PROBLEM CONTROL.TYPE))
       (CONTROL.VALUE.ERROR 3 (ARGUMENT.PROBLEM CONTROL.TYPE))
       (SCOPE.TYPE.ERROR 4 (ARGUMENT.PROBLEM SCOPE.TYPE))
       (SCOPE.VALUE.ERROR 5 (ARGUMENT.PROBLEM SCOPE.TYPE))
       (ACCESS.ERROR 6 (ACCESS.PROBLEM))
       (AUTHENTICATION.ERROR 7 ((AUTHENTICATION . PROBLEM)))
       (CONNECTION.ERROR 8 (CONNECTION.PROBLEM))
       (HANDLE.ERROR 9 (HANDLE.PROBLEM))
       (INSERTION.ERROR 10 (INSERTION.PROBLEM))
       (SERVICE.ERROR 11 (SERVICE.PROBLEM))
       (SESSION.ERROR 12 (SESSION.PROBLEM))
       (SPACE.ERROR 13 (SPACE.PROBLEM))
       (TRANSFER.ERROR 14 (TRANSFER.PROBLEM))
       (UNDEFINED.ERROR 15 (CARDINAL))
       (RANGE.ERROR 16 (ARGUMENT.PROBLEM))))

(COURIERPROGRAM FILING.4 (10 4)
    INHERITS
      (FILING)
    TYPES
      [(SCOPE.TYPE (ENUMERATION (COUNT 0)
                          (DIRECTION 1)
                          (FILTER 2)
                          (DEPTH 3)))
       (SCOPE (CHOICE (COUNT 0 CARDINAL)
                     (DIRECTION 1 DIRECTION)
                     (FILTER 2 FILTER)
```

```
                    (DEPTH 4 CARDINAL)))
       (ACCESS.LIST (RECORD (ENTRIES (SEQUENCE ACCESS.ENTRY))
                            (DEFAULTED BOOLEAN)))
       (ACCESS.ENTRY (RECORD (KEY (CLEARINGHOUSE . NAME))
                             (TYPE (ENUMERATION (INDIVIDUAL 0)
                                                (ALIAS 1)
                                                (GROUP 2)
                                                (-- 3)))
                             (ACCESS UNSPECIFIED])

(DECLARE%: EVAL@COMPILE DONTCOPY

(RPAQQ NSFILINGCONSTANTS
       ((\NSFILING.ALL.ATTRIBUTE.TYPES '(-1))
        (\NSFILING.DEFAULT.TIMEOUT -1)
        (\NSFILING.NULL.FILTER '(ALL))
        (\NSFILING.NULL.FILE.ID '(0 0 0 0 0))
        (\NSFILING.LOWEST.VERSION 0)
        (\NSFILING.HIGHEST.VERSION 65535)
        (\NSFILING.TYPE.BINARY 0)
        (\NSFILING.TYPE.DIRECTORY 1)
        (\NSFILING.TYPE.TEXT 2)))

(DECLARE%: EVAL@COMPILE

(RPAQQ \NSFILING.ALL.ATTRIBUTE.TYPES (-1))

(RPAQQ \NSFILING.DEFAULT.TIMEOUT -1)

(RPAQQ \NSFILING.NULL.FILTER (ALL))

(RPAQQ \NSFILING.NULL.FILE.ID (0 0 0 0 0))

(RPAQQ \NSFILING.LOWEST.VERSION 0)

(RPAQQ \NSFILING.HIGHEST.VERSION 65535)

(RPAQQ \NSFILING.TYPE.BINARY 0)

(RPAQQ \NSFILING.TYPE.DIRECTORY 1)

(RPAQQ \NSFILING.TYPE.TEXT 2)

(CONSTANTS (\NSFILING.ALL.ATTRIBUTE.TYPES '(-1))
       (\NSFILING.DEFAULT.TIMEOUT -1)
       (\NSFILING.NULL.FILTER '(ALL))
       (\NSFILING.NULL.FILE.ID '(0 0 0 0 0))
       (\NSFILING.LOWEST.VERSION 0)
       (\NSFILING.HIGHEST.VERSION 65535)
       (\NSFILING.TYPE.BINARY 0)
       (\NSFILING.TYPE.DIRECTORY 1)
       (\NSFILING.TYPE.TEXT 2))
)

(DECLARE%: EVAL@COMPILE

(ACCESSFNS NSFILINGSTREAM (                                    ; Overlays STREAM.  F1-2 and FW6-8 are used by the bulkdata
                                                               ; device
                           (NSFILING.CONNECTION (fetch F3 of DATUM)
                                  (replace F3 of DATUM with NEWVALUE))
                                                               ; Session on which this stream is open
                           (NSFILING.HANDLE (fetch F4 of DATUM)
                                  (replace F4 of DATUM with NEWVALUE))
                                                               ; Filing HANDLE
                           (NSFILING.NEW.ATTRIBUTES (fetch F5 of DATUM)
                                  (replace F5 of DATUM with NEWVALUE))
                                                               ; For output sequential files, the attributes to install after we write
                                                               ; the file
                           (NSFILING.PAGE.CACHE (fetch F1 of DATUM)
                                  (replace F1 of DATUM with NEWVALUE))
                                                               ; Cache of pages read from server but not yet read by client
                           (NSFILING.SERVER.LENGTH (fetch F2 of DATUM)
                                  (replace F2 of DATUM with NEWVALUE))
                                                               ; For random-access streams, actual length of file on server
                           (NSFILING.LAST.REQUEST (fetch FW6 of DATUM)
                                  (replace FW6 of DATUM with NEWVALUE))
                                                               ; Last page requested to be read or written

                           ))

(DATATYPE FILINGSESSION ((FSLOGINCHANGED FLAG)                 ; True if login info changes for this host
                         (FSREALACTIVITY FLAG)                 ; Set true when there have been non-CONTINUE calls made on
                                                               ; this session
                         (NIL BITS 6)
                         (FSPARSEDNAME POINTER)                ; Canonical NSNAME of server
                         (FSNAMESTRING POINTER)                ; same as a Lisp string
                         (FSADDRESS POINTER)                   ; NSADDRESS of server
```

```
                        (FSPROCESSNAME POINTER)                 ; Courier stream open for this session, or NIL if none
                        (FSSESSIONHANDLE POINTER)               ; Handle for this session
                        (FSSESSIONLOCK POINTER)
                        (FSLASTREALACTIVITYTIMER POINTER)       ; Time of last interesting activity
                        (FSDEVICENAME POINTER)
                        (FSCOURIERSTREAMS POINTER)              ; Courier streams usable by session
                        (FSCACHEDHANDLES POINTER)               ; Zero or more instances of FILINGHANDLE describing handles
                                                                ; we have open in this session
                        (FSLOGINNAME POINTER)                   ; Name under which this session is logged in
                        (FSPROTOCOLNAME POINTER)                ; FILING or OLDFILING
                        (FSPROTOCOLDEF POINTER)                 ; Courier def for FILING.CALL to use
                        (FSSESSIONTIMER POINTER)                ; Time we last did anything at all in this session
                        (FSCONTINUANCE WORD)                    ; How long in msecs we can be idle without having server close
                                                                ; session
                        (FSVERSION WORD)                        ; Version of the protocol in use by this server
                                                                ; Spares
                        (NIL POINTER)
                        (NIL POINTER)
                        (NIL POINTER)))

(DATATYPE FILINGHANDLE ((NSHDIRECTORYP FLAG)                    ; Handle is a directory
                        (NSHWASREAD FLAG)                       ; True if we have read file since we obtained the handle (in which
                                                                ; case read date has been updated)
                        (NSHWASWRITTEN FLAG)
                        (NSHWASMODIFIED FLAG)
                        (NIL BITS 4)
                        (NSHDATUM POINTER)                      ; The file handle datum used in Courier calls
                        (NSHFILEID POINTER)                     ; FILE.ID of file
                        (NSHNAME POINTER)                       ; Full name of the file referenced
                        (NSHPATHNAME POINTER)                   ; Canonical pathname of file
                        (NSHATTRIBUTES POINTER)                 ; Cached attributes
                        (NSHACCESS POINTER)                     ; Current access controls on handle
                        (NSHTIMER POINTER)                      ; Last reference to this handle
                        (NSHBUSYCOUNT WORD)                     ; Number of current users of handle
                        (NIL WORD)
                        (NSHDIRECTORYPATH POINTER)              ; For directories, the list of component dirs
                        (NIL POINTER))
       NSHTIMER _ (SETUPTIMER 0)
       NSHDIRECTORYPATH _ T)

(RECORD NSFILESERVER (NSFSPARSEDNAME . NSFSADDRESSES))

(RECORD NSFILINGDEVICEINFO (NSFILESERVER NSWATCHERPROC NSFILINGLOCK NSFILINGNAME NSRANDOMDEVICE . NSCONNECTIONS)
       )

(RECORD \NSFILING.GENFILESTATE (CURRENTINFO NSCONNECTION NSGENERATOR NSFILTER NSIGNOREDIRECTORIES NSBULKSTREAM))

(RECORD NSFILINGPARSE (NSDIRECTORIES NSROOTNAME NSVERSION NSDIRECTORYP NSHASPERIOD))

(RECORD NSPAGECACHE (NSPSIZE . NSPHEADER)
       (RECORD NSPHEADER (NSPTAIL . NSPBUFFERS)))
)

(/DECLAREDATATYPE 'FILINGSESSION
       '(FLAG FLAG (BITS 6)
              POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
              POINTER POINTER WORD WORD POINTER POINTER POINTER)
       ;; ---field descriptor list elided by lister---
       '38)

(/DECLAREDATATYPE 'FILINGHANDLE '(FLAG FLAG FLAG FLAG (BITS 4)
                                      POINTER POINTER POINTER POINTER POINTER POINTER POINTER WORD WORD POINTER
                                      POINTER)
       ;; ---field descriptor list elided by lister---
       '22)

(DECLARE%: EVAL@COMPILE

(PUTPROPS WITHOUT.SESSION.MONITOR MACRO [(SESSION . FORMS)
                                        (LET ((LOCK (fetch FSSESSIONLOCK of SESSION)))
                                             (DECLARE (LOCALVARS LOCK))
                                             (RELEASE.MONITORLOCK LOCK)
                                             (PROG1 (PROGN . FORMS)
                                                    (OBTAIN.MONITORLOCK LOCK])
)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS \NSFILING.CONNECTIONS \NSFILING.DEVICE \NSFILING.NULL.HANDLE \NSFILING.ATTRIBUTES
       \LISP.TO.NSFILING.ATTRIBUTES \NSFILING.USEFUL.ATTRIBUTE.TYPES \NSFILING.PROGRAM.NAME
       \NSFILING.ACTIVE.SESSIONS FILING.CACHE.LIMIT *NSFILING-PAGE-CACHE-INCREMENT* *NSFILING-PAGE-CACHE-LIMIT*
       *NSFILING-RANDOM-ACCESS* *NSFILING-SESSION-TIMEOUT* \NSRANDOM.CHECK.CACHE \NSFILING.PROTECTION.BITS
       \FILEDEVICES)
)
```

```
(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)
)

(FILESLOAD (SOURCE)
       SPPDECLS)

(FILESLOAD (LOADCOMP)
       COURIER)
)

(/DECLAREDATATYPE 'FILINGSESSION
       '(FLAG FLAG (BITS 6)
              POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
              POINTER POINTER WORD WORD POINTER POINTER POINTER)
       ;; ---field descriptor list elided by lister---
       '38)

(/DECLAREDATATYPE 'FILINGHANDLE '(FLAG FLAG FLAG FLAG (BITS 4)
                                    POINTER POINTER POINTER POINTER POINTER POINTER POINTER WORD WORD POINTER
                                    POINTER)
       ;; ---field descriptor list elided by lister---
       '22)

(DEFINEQ
```

(\FILINGSESSION.DEFPRINT

```
  [LAMBDA (SESSION STREAM)                                    ; Edited  1-Jun-87 16:58 by bvm:
    (COND
       ((AND COURIERTRACEFILE (TYPENAMEP COURIERTRACEFILE 'WINDOW)
             (EQ (ffetch (WINDOW DSP) of COURIERTRACEFILE)
                 STREAM))                                     ; Want it curt in trace output
        NIL)
       (T (\DEFPRINT.BY.NAME SESSION STREAM (fetch FSNAMESTRING of SESSION)
                 "Filing Session on"])
```

(\FILINGHANDLE.DEFPRINT

```
  [LAMBDA (HANDLE STREAM)                                     ; Edited 15-May-87 17:10 by bvm:
    (\DEFPRINT.BY.NAME HANDLE STREAM (OR (fetch NSHNAME of HANDLE)
                                         (fetch NSHPATHNAME of HANDLE))
          "Filing Handle on"])

)

(DEFINEQ
```

(\GET.FILING.ATTRIBUTE

```
  [LAMBDA (STREAM PROGRAM)                                    (* bvm%: "25-Jul-86 16:48")
    ;; Reads a filing attribute value pair from STREAM, returning a list of two elements, (ATTR VALUE);  if the attribute is not a known attribute, ATTR
    ;; is an integer and VALUE is a sequence of unspecified
    (bind (ATTR _ (COURIER.READ STREAM NIL 'LONGCARDINAL))
          VALUE for X in \NSFILING.ATTRIBUTES when (EQ (CADR X)
                                                       ATTR)
       do [RETURN (CONS (CAR X)
                        (COND
                           ((EQ (\WIN STREAM)
                                0)                           ; sequence count zero means no value is here
                            NIL)
                           (T                                ; Ignore sequence count, read as known kind of data
                              (LIST (COURIER.READ STREAM PROGRAM (CADDR X]
       finally                                               ; ATTR not recognized
              (RETURN (LIST ATTR (COURIER.READ.SEQUENCE STREAM NIL 'UNSPECIFIED])
```

(\PUT.FILING.ATTRIBUTE

```
  [LAMBDA (STREAM ITEM PROGRAM)                               (* bvm%: "15-Jan-85 16:29")
```

;;; Writes a filing attribute value pair to STREAM. ITEM is a list of two elements (ATTR VALUE)

```
    (PROG ((ATTR (CAR ITEM))
           (VALUE (CADR ITEM))
           VALUETYPE)
          [COND
             ((NOT (FIXP ATTR))
              (for X in \NSFILING.ATTRIBUTES when (EQ (CAR X)
                                                      ATTR)
                 do (SETQ ATTR (CADR X))
                    (SETQ VALUETYPE (CADDR X))
                    (RETURN)
                 finally (ERROR "Unknown Filing attribute" ATTR]
          (COURIER.WRITE STREAM ATTR NIL 'LONGCARDINAL)
          (COND
```

```
            (VALUETYPE (COURIER.WRITE.SEQUENCE.UNSPECIFIED STREAM VALUE PROGRAM VALUETYPE))
            (T (COURIER.WRITE.SEQUENCE STREAM VALUE PROGRAM 'UNSPECIFIED])
```

(\GET.SESSION.HANDLE

```
  [LAMBDA (STREAM PROGRAM)                                    ; Edited 1-Jun-87 15:52 by bvm:
    ;; Read an object of type Filing.Session, which consists of a token (array 2 unspecified) and a verifier.
    (CONS (COURIER.READ STREAM NIL 'UNSPECIFIED)
          (CONS (COURIER.READ STREAM NIL 'UNSPECIFIED)
                (COURIER.READ STREAM 'AUTHENTICATION 'VERIFIER])
```

(\PUT.SESSION.HANDLE

```
  [LAMBDA (STREAM ITEM PROGRAM)                               ; Edited 1-Jun-87 15:52 by bvm:
    ;; Write a session handle.  This is where we can stick hook to increment verifier when we start using strong authentication.  Handle = (token token .
    ;; verifier).
    (LET [(HANDLE (OR (LISTP ITEM)
                      [LISTP (ffetch FSSESSIONHANDLE of (\DTEST ITEM 'FILINGSESSION]
                      (ERROR "Attempt to use obsolete session" ITEM]
          (COURIER.WRITE STREAM (pop HANDLE)
                 NIL
                 'UNSPECIFIED)
          (COURIER.WRITE STREAM (pop HANDLE)
                 NIL
                 'UNSPECIFIED)
          (COURIER.WRITE STREAM HANDLE 'AUTHENTICATION 'VERIFIER)
          ITEM])

)

(PUTPROPS FILING.SESSION COURIERDEF (\GET.SESSION.HANDLE \PUT.SESSION.HANDLE))

(PUTPROPS FILING.ATTRIBUTE COURIERDEF (\GET.FILING.ATTRIBUTE \PUT.FILING.ATTRIBUTE))

(DECLARE%: EVAL@COMPILE DOCOPY

(RPAQQ \NSFILING.NULL.HANDLE (0 0))

(RPAQQ \NSFILING.PROTECTION.BITS ((READ . 16)
                                  (WRITE . 8)
                                  (DELETE . 1)
                                  (CREATE . 2)
                                  (MODIFY . 4)))

(RPAQQ \NSFILING.ATTRIBUTES
       ((CHECKSUM 0 CARDINAL)
        (CHILDREN.UNIQUELY.NAMED 1 BOOLEAN)
        (CREATED.BY 2 USER)
        (CREATED.ON 3 TIME)
        (FILE.ID 4 FILE.ID)
        (IS.DIRECTORY 5 BOOLEAN)
        (IS.TEMPORARY 6 BOOLEAN)
        (MODIFIED.BY 7 USER)
        (MODIFIED.ON 8 TIME)
        (NAME 9 STRING)
        (NUMBER.OF.CHILDREN 10 CARDINAL)
        (ORDERING 11 ORDERING)
        (PARENT.ID 12 FILE.ID)
        (POSITION 13 (SEQUENCE UNSPECIFIED))
        (READ.BY 14 USER)
        (READ.ON 15 TIME)
        (SIZE.IN.BYTES 16 LONGCARDINAL)
        (FILE.TYPE 17 LONGCARDINAL)
        (VERSION 18 CARDINAL)
        (ACCESS.LIST 19 ACCESS.LIST)
        (DEFAULT.ACCESS.LIST 20 ACCESS.LIST)
        (PATHNAME 21 STRING)
        (BACKED.UP.ON 23 TIME)
        (FILED.BY 24 USER)
        (FILED.ON 25 TIME)
        (STORED.SIZE 26 LONGCARDINAL)
        (SUBTREE.SIZE 27 LONGCARDINAL)
        (SUBTREE.SIZE.LIMIT 28 LONGCARDINAL)
        (OWNER 4351 STRING)))

(RPAQQ \LISP.TO.NSFILING.ATTRIBUTES
       ((IWRITEDATE MODIFIED.ON)
        (IREADDATE READ.ON)
        (ICREATIONDATE CREATED.ON)
        (CREATIONDATE CREATED.ON)
        (READDATE READ.ON)
        (WRITEDATE MODIFIED.ON)
        (LENGTH SIZE.IN.BYTES)
        (AUTHOR CREATED.BY)
        (READER READ.BY)
```

```
        (PROTECTION ACCESS.LIST)
        (SIZE SIZE.IN.BYTES)
        (TYPE FILE.TYPE)
        (FILETYPE FILE.TYPE)))

(RPAQ \NSFILING.USEFUL.ATTRIBUTE.TYPES (\FILING.ATTRIBUTE.TYPE.SEQUENCE '(CREATED.ON FILE.ID IS.DIRECTORY
                                                                           PATHNAME SIZE.IN.BYTES FILE.TYPE
                                                                           VERSION)))
)

(RPAQ? FILING.CACHE.LIMIT 6)

(RPAQ? NSFILING.SHOW.STATUS T)

(RPAQ? FILING.ENUMERATION.DEPTH T)

(RPAQ? \NSFILING.LOCK (CREATE.MONITORLOCK 'NSFILING))

(RPAQ? \NSFILING.PROGRAM.NAME 'FILING)

(RPAQ? \NSFILING.ACTIVE.SESSIONS )

(RPAQ? *NSFILING-RANDOM-ACCESS* T)

(RPAQ? *NSFILING-PAGE-CACHE-LIMIT* 8)

(RPAQ? *NSFILING-PAGE-CACHE-INCREMENT* 4)

(RPAQ? *NSFILING-SESSION-TIMEOUT* '(900 . 21600))

(RPAQ? \NSRANDOM.CHECK.CACHE )
```

;; Connection maintenance

```
(DEFINEQ
```

(\GETFILINGCONNECTION

```
  [LAMBDA (DEVICE OLDSTREAM NOLOCK)                          ; Edited 18-May-87 17:53 by bvm:
```

;;; Find an existing session on this fileserver or log in a new one. Returns the session, after obtaining its monitor lock. Caller must have a RESETLST

```
    (LET* [(DEVINFO (fetch DEVICEINFO of DEVICE))
           (SESSION (WITH.MONITOR (fetch NSFILINGLOCK of DEVINFO)
                        (bind SESSION while (SETQ SESSION (CAR (fetch NSCONNECTIONS of DEVINFO)))
                           do                                ; Awkward control structure because of DREMOVE
                               [COND
                                  ((WITH.MONITOR (fetch FSSESSIONLOCK of SESSION)
                                                 (\VALID.FILING.CONNECTIONP SESSION))
                                                             ; If good, returned session. If bad, returned possibly an open
                                                             ; courier stream
                                   (RETURN SESSION))
                                  (T (SETQ OLDSTREAM (\NSFILING.DISCARD.SESSION SESSION DEVICE (NULL OLDSTREAM]
                            finally (RETURN (\NSFILING.LOGIN DEVINFO DEVICE OLDSTREAM))))]
          (COND
             (SESSION (COND
                         ((NOT NOLOCK)                       ; Grab lock here outside of our own WITH.MONITOR.
                                                             ; Unwindsave info goes on caller's reset
                          (OBTAIN.MONITORLOCK (fetch FSSESSIONLOCK of SESSION)
                                 NIL T)))
                      SESSION])
```

(\NSFILING.GET.NEW.SESSION

```
  [LAMBDA (OLDSESSION DEVICE NOLOCK)                         ; Edited 22-May-87 14:42 by bvm:
```

;; Called when OLDSESSION has encountered a session error (TokenInvalid). Discards knowledge of OLDSESSION and establishes a new one.
;; Unless NOLOCK is true, a lock is obtained on the new session (caller must have RESETLST).

```
    (\GETFILINGCONNECTION DEVICE (\NSFILING.DISCARD.SESSION OLDSESSION DEVICE T)
           NOLOCK])
```

(\NSFILING.GET.STREAM

```
  [LAMBDA (CONNECTION KEEPSTREAM)                            ; Edited  9-Jun-87 15:41 by bvm:
```

;;; Get a Courier stream for CONNECTION and return it. If KEEPSTREAM is true, we want the stream to persist after the enclosing RESETLST exits,
;;; else we release the stream on its exit

```
    (PROG [(STREAMPAIR (find PAIR in (fetch FSCOURIERSTREAMS of CONNECTION) suchthat (NULL (CDR PAIR]
          (COND
             (STREAMPAIR (RPLACD STREAMPAIR T))
             [(SETQ STREAMPAIR (\NSFILING.COURIER.OPEN (fetch FSADDRESS of CONNECTION)
                                                       (fetch FSPROCESSNAME of CONNECTION)))
              (push (fetch FSCOURIERSTREAMS of CONNECTION)
                    (SETQ STREAMPAIR (CONS STREAMPAIR T]
             (T (RETURN NIL)))
          (RESETSAVE NIL (LIST [FUNCTION (LAMBDA (CONNECTION PAIR KEEPSTREAM)
                                           (COND
```

```
                                        [RESETSTATE (SPP.CLOSE (CAR PAIR)
                                                           T)
                                               (replace FSCOURIERSTREAMS of CONNECTION
                                                  with (DREMOVE PAIR (fetch FSCOURIERSTREAMS of CONNECTION]
                                        ((NOT KEEPSTREAM)
                                         (RPLACD PAIR NIL]
                                 CONNECTION STREAMPAIR KEEPSTREAM))
        (RETURN STREAMPAIR])
```

(\NSFILING.COURIER.OPEN

```
[LAMBDA (ADDRESS NAME)                                          (* bvm%: "11-Dec-85 12:57")
  (COURIER.OPEN ADDRESS NIL T NAME (FUNCTION \NSFILING.WHENCLOSED)
         (CONSTANT (LIST 'ERRORHANDLER (FUNCTION \NSFILING.ERRORHANDLER])
```

(\NSFILING.CLOSE.BULKSTREAM

```
[LAMBDA (CONNECTION STREAM)                                     ; Edited 20-Nov-87 18:47 by bvm:
  (COND
     ((AND STREAM (OPENP STREAM))
      (CLOSEF STREAM)
      (\NSFILING.RELEASE.BULKSTREAM CONNECTION STREAM RESETSTATE])
```

(\NSFILING.RELEASE.BULKSTREAM

```
[LAMBDA (CONNECTION STREAM ABORT?)                              (* bvm%: "11-Dec-85 14:42")
  (LET ((STREAMS (fetch FSCOURIERSTREAMS of CONNECTION)))
       (for PAIR in STREAMS when (EQ (CDR PAIR)
                                     STREAM)
          do (COND
                (ABORT?                                         ; Unknown state, bag it
                       (SPP.CLOSE (CAR PAIR)
                                  T)
                       (replace FSCOURIERSTREAMS of CONNECTION with (DREMOVE PAIR STREAMS)))
                (T                                              ; Stream now free
                   (RPLACD PAIR NIL)))
             (RETURN])
```

(FILING.CALL

```
[LAMBDA ARGS                                                    ; Edited  5-Aug-87 12:39 by bvm:
```

;; Call a FILING procedure. procedure, in a style similar to COURIER.CALL --- (FILING.CALL session procedure-name arg1 ... argN) --- Returns
;; the result of the remote procedure, or a list of such results if it returns more than one. A single flag NoErrorFlg can be optionally appended to the
;; arglist -- If NoErrorFlg is NOERROR, return NIL if the Courier program aborts with an error; if RETURNERRORS, then return an expression
;; (ERROR ERRNAME . args) on error.

;; Copied from COURIER.CALL

```
(PROG (SESSION PROCEDURE PROGRAM STREAM NARGS ARGLIST NOERRORFLG PGMDEF PROCDEF ARGTYPES KEEPSTREAM
             ABSOLUTELY-NO-ERROR)
       [COND
          ((< ARGS 2)
           (RETURN (ERROR "Malformed FILING.CALL"]
       [if (NULL (SETQ SESSION (ARG ARGS 1)))
           then                                                 ; session killed, don't even try
            (RETURN '(ERROR SESSION.ERROR TokenInvalid]
       (SETQ PGMDEF (fetch FSPROTOCOLDEF of SESSION))
       (SETQ PROCDEF (\GET.COURIER.DEFINITION (SETQ PROGRAM (fetch FSPROTOCOLNAME of SESSION))
                           (SETQ PROCEDURE (ARG ARGS 2))
                           'PROCEDURES PGMDEF))
       [SETQ NARGS (LENGTH (SETQ ARGTYPES (fetch (COURIERFN ARGS) of PROCDEF]
       [OR (SELECTQ (- ARGS NARGS)
                (2                                              ; Exactly right
                   T)
                ((3 4)                                          ; Extra arg is errorflg
                     (AND (SELECTQ (SETQ NOERRORFLG (ARG ARGS (+ NARGS 3)))
                                   (NOERROR                     ; Caller wants not to hassle with errors, but we always want to
                                                                ; see session errors
                                            (SETQ NOERRORFLG 'RETURNERRORS)
                                            (SETQ ABSOLUTELY-NO-ERROR T))
                                   ((NOERROR RETURNERRORS NIL)
                                      T)
                                   NIL)
                          (COND
                             [(EQ (- ARGS NARGS)
                                  4)
                              (SETQ KEEPSTREAM (EQ (ARG ARGS (+ NARGS 4))
                                                   'KEEPSTREAM]
                             (T T))))
                NIL)
           (RETURN (ERROR "Wrong number of arguments to Courier procedure" (CONS PROGRAM PROCEDURE]
       (SETQ ARGLIST (for I from 3 to (+ NARGS 2) collect (ARG ARGS I)))
       (RETURN (WITH.MONITOR (fetch FSSESSIONLOCK of SESSION)   ; Note: implicit RESETLST
                   [PROG ((FAILED 0)
                          STREAM RESULT)
                     NEWSTREAM
                         [COND
                            [(NOT (LISTP (fetch FSSESSIONHANDLE of SESSION)))
```

```
                                                    ; Session is dead, don't even try the call
                (RETURN '(ERROR SESSION.ERROR TokenInvalid]
               ((NULL (SETQ STREAM (\NSFILING.GET.STREAM SESSION KEEPSTREAM)))
                (COND
                   (ABSOLUTELY-NO-ERROR (RETURN NIL))
                   (T (COND
                         ((EQ (add FAILED 1)
                              2)                    ; Don't complain the first time--it seems like it often takes a while
                                                    ; to wake up a sleepy server.  Perhaps should adjust this in
                                                    ; SPP.OPEN.
                          (PRINTOUT PROMPTWINDOW T "No response from " (fetch FSNAMESTRING
                                                                          of SESSION)
                                 ";" " will keep trying.")))
                      (GO NEWSTREAM]
          (SETQ RESULT (COURIER.EXECUTE.CALL (CAR STREAM)
                               PROGRAM PGMDEF PROCEDURE PROCDEF ARGLIST ARGTYPES NOERRORFLG))
          [COND
             ((EQ RESULT 'STREAM.LOST)
              (GO NEWSTREAM))
             ((AND (LISTP RESULT)
                   (EQ (CAR RESULT)
                       'ERROR)
                   (SELECTQ (CADR RESULT)
                        (SESSION.ERROR              ; Dead session
                           T)
                        (REJECT (SELECTQ (CAADDR RESULT)
                                     ((NoSuchService WrongVersionOfService)

                        ;; Server not responding to Filing?  Could happen if server crashed and has just come back.  In any
                        ;; case, our old session is clearly dead--we masquerade here as session error and let LOGIN worry
                        ;; about proceeding.

                                 (SETQ RESULT '(ERROR SESSION.ERROR TokenInvalid)))
                                NIL))
                    NIL))
              ;; Session is dead, don't let anybody even think about using it again.  If caller is clever, however, he may
              ;; reuse the stream to login afresh.
              (replace FSSESSIONHANDLE of SESSION with :CLOSED))
             (T (COND
                   ((NEQ PROCEDURE 'CONTINUE)  ; Note real activity
                    (replace FSREALACTIVITY of SESSION with T))
                   ((fetch FSREALACTIVITY of SESSION)
                                                    ; transfer activity timer to real timer
                    (\BLT (OR (fetch FSLASTREALACTIVITYTIMER of SESSION)
                              (replace FSLASTREALACTIVITYTIMER of SESSION with (\CREATECELL
                                                                                \FIXP)))
                          (fetch FSSESSIONTIMER of SESSION)
                          WORDSPERCELL)
                    (replace FSREALACTIVITY of SESSION with NIL)))
                (\DAYTIME0 (fetch FSSESSIONTIMER of SESSION))
                                                    ; Note time of last activity
                (COND
                   (KEEPSTREAM (RPLACD STREAM (COND
                                                 ((TYPENAMEP RESULT 'STREAM)
                                                    ; Save bulk stream for later linkup
                                                  RESULT)
                                                 (T
                                                    ; Were expecting bulk stream but failed, so release main stream
                                                  NIL]
          (RETURN (COND
                     ((AND ABSOLUTELY-NO-ERROR (LISTP RESULT)
                           (EQ (CAR RESULT)
                               'ERROR))             ; Manually suppress this error.
                      NIL)
                     (T RESULT])])
```

(\NSFILING.LOGIN

```
 [LAMBDA (DEVINFO DEVICE STREAM)                    ; Edited 9-Feb-88 15:58 by bvm:
   (RESETLST
       [RESETSAVE NIL (LIST (FUNCTION (LAMBDA NIL               ; Close any open stream on error
                                (AND STREAM RESETSTATE (SPP.CLOSE STREAM]
       (LET ((FILESERVER (fetch NSFILESERVER of DEVINFO))
             (PROCNAME (fetch NSFILINGNAME of DEVINFO))
             (PROGRAM \NSFILING.PROGRAM.NAME)
             ADDRESS SERVERNAME SERVERNSNAME SESSIONHANDLE SESSION CREDENTIALS NEEDLOGIN PROBLEM OLDPROBLEM
             LOGINNAME RANDEVICE)
            (SETQ SERVERNAME (MKATOM (NSNAME.TO.STRING (SETQ SERVERNSNAME (fetch NSFSPARSEDNAME of FILESERVER))
                                     T)))
            (SETQ ADDRESS (CAR (fetch NSFSADDRESSES of FILESERVER)))
            (COND
               ([when [COND
                         ([NOT (SETQ CREDENTIALS (\INTERNAL/GETPASSWORD SERVERNAME NEEDLOGIN NIL
                                                    [COND
                                                       (NEEDLOGIN (PROG1 (SELECTQ NEEDLOGIN
                                                                            (VerifierInvalid
                                                                                 "Incorrect Password")
```

```
                                                                    (CredentialsInvalid
                                                                        "Invalid User Name")
                                                                    (CONCAT "Login failed -- "
                                                                            NEEDLOGIN))
                                                                (SETQ NEEDLOGIN NIL]
                                                NIL
                                                'NS]    ; User aborted
                    (RETURN NIL))
                   ([NOT (OR STREAM (SETQ STREAM (\NSFILING.COURIER.OPEN ADDRESS PROCNAME]
                                                        ; No response
                    (SETQ PROBLEM T))
                   (T (SETQ LOGINNAME (CAR CREDENTIALS))
                      (SETQ CREDENTIALS (NS.MAKE.SIMPLE.CREDENTIALS CREDENTIALS))
                      (SETQ SESSIONHANDLE (COURIER.CALL STREAM PROGRAM 'LOGON SERVERNSNAME (CAR
                                                                                        CREDENTIALS
                                                                                          )
                                                    (CDR CREDENTIALS)
                                                    'RETURNERRORS))
                      (COND
                         ((EQ SESSIONHANDLE 'STREAM.LOST) ; Stream was idle too long before we made that call, so toss it
                                                        ; and get a new one.
                          (SETQ STREAM NIL))
                         ((NULL SESSIONHANDLE)          ; Shouldn't happen, treat as no response
                          (SETQ PROBLEM T))
                         ((NEQ (CAR SESSIONHANDLE)
                               'ERROR)                  ; Success
                          (RETURN SESSIONHANDLE))
                         (T (SELECTQ (CADR SESSIONHANDLE)
                                 (REJECT                ; Can't handle this call
                                         (SELECTQ (CAR (CADDR SESSIONHANDLE))
                                              (WrongVersionOfService
                                                   (COND
                                                      ((EQ PROGRAM 'FILING)
                                                       (SETQ PROGRAM 'FILING.4)
                                                                ; Quietly try older version next time around
                                                       NIL)
                                                      (T        ; Doesn't run any version we talk
                                                         T)))
                                              (NoSuchService
                                                                ; Can happen when you boot a file server.  Keep trying
                                                   (SETQ PROBLEM 'NoSuchService))
                                              T))
                                 (AUTHENTICATION.ERROR
                                      (SETQ NEEDLOGIN (CADDR SESSIONHANDLE))
                                                                ; Login incorrect, prompt next time around
                                      NIL)
                                 (SERVICE.ERROR
                                      (SELECTQ (SETQ PROBLEM (CADDR SESSIONHANDLE))
                                           ((CannotAuthenticate ServiceFull)
                                                                ; hopefully transient problems
                                            T)
                                           (UnknownService
                                                                ; No service by that name at this node.  This is quite transient in
                                                                ; the case where the server was just booted
                                            (if (NEQ OLDPROBLEM 'NoSuchService)
                                                then (SETQ PROBLEM NIL))
                                            T)
                                           (PROGN (SETQ PROBLEM NIL)
                                                                ; Let other problems cause a break
                                                  T)))
                                 T]
                   do                                   ; Some sort of problem encountered.  PROBLEM set non-nil if
                                                        ; it's worthwhile to keep trying, else an unexpected problem is
                                                        ; stored in SESSIONHANDLE
                   (COND
                      ((NULL PROBLEM)
                       (SPP.CLOSE STREAM)
                       (SETQ STREAM NIL)
                       (CL:CERROR "Try again to connect" "Error while logging on to ~A: ~A~%(Type OK to try
                              again)" SERVERNAME (CDR SESSIONHANDLE)))
                      (T (COND
                            ((NEQ PROBLEM OLDPROBLEM)
                             (PRINTOUT PROMPTWINDOW T "Can't connect to " SERVERNAME " because: "
                                    (SELECTQ (SETQ OLDPROBLEM PROBLEM)
                                         (T "No response")
                                         (NoSuchService
                                              "File Service not running")
                                         OLDPROBLEM)
                                    ";" " will keep trying.")))
                         (SETQ PROBLEM NIL)
                         (DISMISS (COND
                                     ((EQ OLDPROBLEM T)     ; No explicit response, just try soon
                                      5000)
                                     (T                     ; It's likely to take a while to get going
                                        30000]
;; Succeeded in logging in
```

```
                (if (AND OLDPROBLEM (NEQ OLDPROBLEM T))
                    then                                          ; Let us know when successful
                         (PRINTOUT PROMPTWINDOW T "Got connection to " SERVERNAME))
                (push \NSFILING.ACTIVE.SESSIONS
                      (SETQ SESSION
                       (create FILINGSESSION
                              FSADDRESS _ ADDRESS
                              FSPARSEDNAME _ SERVERNSNAME
                              FSNAMESTRING _ SERVERNAME
                              FSPROCESSNAME _ PROCNAME
                              FSCOURIERSTREAMS _ (LIST (CONS STREAM))
                              FSSESSIONHANDLE _ SESSIONHANDLE
                              FSPROTOCOLNAME _ PROGRAM
                              FSDEVICENAME _ (fetch (FDEV DEVICENAME) of DEVICE)
                              FSPROTOCOLDEF _ (\GET.COURIERPROGRAM PROGRAM)
                              FSSESSIONLOCK _ (CREATE.MONITORLOCK SERVERNAME)
                              FSSESSIONTIMER _ (\CREATECELL \FIXP)
                              FSLOGINNAME _ LOGINNAME)))
                (\NSFILING.SET.CONTINUANCE SESSION)
                (push (fetch NSCONNECTIONS of DEVINFO)
                      SESSION)
                (COND
                   ((AND (EQ PROGRAM 'FILING)
                         (NOT (fetch NSRANDOMDEVICE of DEVINFO)))
                                                                  ; Create a second device to use for random-access streams.
                                                                  ; This is an invisible device, so only needs methods for things
                                                                  ; you can do to open streams
                    [replace NSRANDOMDEVICE of DEVINFO
                       with (SETQ RANDEVICE (\MAKE.PMAP.DEVICE (create FDEV
                                                                       DEVICENAME _ (fetch FSNAMESTRING
                                                                                           of SESSION)
                                                                       OPENFILE _ (FUNCTION \NSRANDOM.OPENFILE)
                                                                       REOPENFILE _ (FUNCTION NILL)
                                                                       CLOSEFILE _ (FUNCTION \NSRANDOM.CLOSEFILE)
                                                                       GETFILEINFO _ (FUNCTION
                                                                                      \NSFILING.GETFILEINFO)
                                                                       SETFILEINFO _ (FUNCTION
                                                                                      \NSFILING.SETFILEINFO)
                                                                       REGISTERFILE _ (FUNCTION NILL)
                                                                       UNREGISTERFILE _ (FUNCTION NILL)
                                                                       READPAGES _ (FUNCTION \NSRANDOM.READPAGES)
                                                                       WRITEPAGES _ (FUNCTION
                                                                                     \NSRANDOM.WRITEPAGES)
                                                                       TRUNCATEFILE _ (FUNCTION
                                                                                       \NSRANDOM.TRUNCATEFILE)
                                                                       DEVICEINFO _ DEVICE
                                                                       REMOTEP _ T
                                                                       SUBDIRECTORIES _ T]
                    (replace SETEOFPTR of RANDEVICE with (FUNCTION \NSRANDOM.SETEOFPTR))
                                                                  ; Have to do this after \make.pmap.device

                    ))
                SESSION)))])
```

(\NSFILING.AFTER.LOGIN

```
  [LAMBDA (HOST USER)                                             (* bvm%: "31-Jan-86 17:45")
    (for SESSION in \NSFILING.ACTIVE.SESSIONS when (OR (NULL HOST)
                                                       (STRING-EQUAL HOST (fetch FSNAMESTRING of SESSION)))
       do                                                         ; Note that the login has changed for this host
        (replace FSLOGINCHANGED of SESSION with T])
```

(\NSFILING.SET.CONTINUANCE

```
  [LAMBDA (SESSION)                                               ; Edited  5-Jun-87 18:11 by bvm:
    (LET [(SECONDS (FILING.CALL SESSION 'CONTINUE SESSION 'RETURNERRORS]
         (COND
            ((FIXP SECONDS)
             ;; Continue value is number of seconds we can be idle.  Take 3/4 of what the server says, just to be conservative
             (replace FSCONTINUANCE of SESSION with (IMIN (IQUOTIENT (ITIMES SECONDS 3)
                                                                     4)
                                                          MAX.SMALLP))
             T])
```

(\NSFILING.LOGOUT

```
  [LAMBDA (SESSION)                                               ; Edited  5-Jun-87 17:54 by bvm:
    (FILING.CALL SESSION 'LOGOFF SESSION 'NOERROR])
```

(\NSFILING.DISCARD.SESSION

```
  [LAMBDA (SESSION DEVICE KEEPSTREAM)                             ; Edited  2-Jun-87 17:55 by bvm:
    ;; Called when SESSION is known to be dead.  If KEEPSTREAM is true, we return some active stream, if any, otherwise all streams are closed.
    (SETQ \NSFILING.ACTIVE.SESSIONS (DREMOVE SESSION \NSFILING.ACTIVE.SESSIONS))
    (change (fetch NSCONNECTIONS of (fetch DEVICEINFO of DEVICE))
```

```
        (DREMOVE SESSION DATUM))
    (PROG1 (for PAIR in (fetch FSCOURIERSTREAMS of SESSION) bind KEPT do [COND
                                                        ((AND KEEPSTREAM (NULL KEPT)
                                                              (NULL (CDR PAIR)))
                                                         ; Keep this stream for caller
                                                          (SETQ KEPT (CAR PAIR)))
                                                        (T (CLOSEF (CAR PAIR]
           finally (replace FSCOURIERSTREAMS of SESSION with NIL)
                   (RETURN KEPT))
     (replace FSSESSIONHANDLE of SESSION with :CLOSED)          ; Let no one be tempted to use it again.
     ;; Finally, release lock if we have it.  This avoids deadlock in the case where we try to get a new session, but somebody before us is already
     ;; inside \GETFILINGCONNECTION holding on to this session lock.
     (RELEASE.MONITORLOCK (fetch FSSESSIONLOCK of SESSION))))])
```

(\VALID.FILING.CONNECTIONP

```
[LAMBDA (SESSION)                                               ; Edited  5-Jun-87 18:11 by bvm:
  ;; true if this is a useable connection
  (COND
     ([AND (fetch FSLOGINCHANGED of SESSION)
           (NOT (STRING-EQUAL (CAR (\INTERNAL/GETPASSWORD (fetch FSNAMESTRING of SESSION)))
                      (fetch FSLOGINNAME of SESSION]            ; Want new session because credentials changed
      (COND
         ((NOT (for PAIR in (fetch FSCOURIERSTREAMS of SESSION) thereis (CDR PAIR)))
          ;; Can't do this if someone is using the session!
          (\NSFILING.LOGOUT SESSION)))
      NIL)
     ((\SECONDSCLOCKGREATERP (fetch FSSESSIONTIMER of SESSION)
             (fetch FSCONTINUANCE of SESSION))                  ; Our conservative timer has expired, but try to use session
                                                                ; anyway
      (BLOCK)
      (BLOCK)                                                   ; Let spp process run, possibly purge streams
      (COND
         ((\NSFILING.SET.CONTINUANCE SESSION)
          SESSION)))
     (T                                                         ; Timer not expired, so we're safe, it says here
        SESSION])
```

(\NSFILING.CLOSE.CONNECTIONS

```
[LAMBDA (DEVICE ABORT?)                                         ; Edited 30-Nov-87 13:18 by bvm:
  ;; ABORT? = {NIL | :TEST | :ABORT}, meaning {do logout | logout if we haven't timed out | never bother logging out}.
  (RESETLST
      (OBTAIN.MONITORLOCK (fetch NSFILINGLOCK of (fetch (FDEV DEVICEINFO) of DEVICE))
             (EQ ABORT? :ABORT)
             T)                                                 ; Note that if ABORT = :ABORT we don't ever wait for the
                                                                ; lock--we're probably screwed anyway if some other process
                                                                ; has the session lock at the instant the SAVEVM, etc.
                                                                ; happened.
      (LET ((DEVINFO (fetch (FDEV DEVICEINFO) of DEVICE))
            SESSION)
           (if (fetch NSWATCHERPROC of DEVINFO)
               then                                             ; Don't need a watcher any more
                    (DEL.PROCESS (fetch NSWATCHERPROC of DEVINFO)))
           (while (SETQ SESSION (CAR (fetch NSCONNECTIONS of DEVINFO)))
              do (COND
                    ([SELECTQ ABORT?
                          (NIL T)
                          (:ABORT                               ; don't bother trying to LOGOUT, we know it's futile
                                  NIL)
                          (PROGN                                ; Assume session is timed out if session timer expired much
                                                                ; longer ago than stated continuance.
                                 (NOT (\SECONDSCLOCKGREATERP (fetch FSSESSIONTIMER of SESSION)
                                                             (TIMES 2 (fetch FSCONTINUANCE of SESSION]
                     (for PAIR in (APPEND (fetch FSCOURIERSTREAMS of SESSION)) when (CDR PAIR)
                        do                                      ; Someone has a courier stream open on this session, e.g., a
                                                                ; bulk data retrieve or write.  If we try to LOGOUT now we will
                                                                ; hang, so bash the stream
                           (SPP.CLOSE (CAR PAIR)
                                  T)
                           (BLOCK))
                     (\NSFILING.LOGOUT SESSION)))
                 (\NSFILING.DISCARD.SESSION SESSION DEVICE)))
      (for STREAM in (fetch (FDEV OPENFILELST) of DEVICE) do    ; invalidate stream's sessions
                                                          (replace NSFILING.CONNECTION of STREAM with NIL)))])
```

(BREAK.NSFILING.CONNECTION

```
[LAMBDA (HOST DEVICE)                                           ; Edited 26-May-87 17:06 by bvm:
  (COND
     ((EQ HOST T)
      (for DEV in \FILEDEVICES when (AND (EQ (fetch (FDEV OPENFILE) of DEV)
                                             (FUNCTION \NSFILING.OPENFILE))
```

```
                                        (fetch NSCONNECTIONS of (fetch (FDEV DEVICEINFO) of DEV)))
          collect (\NSFILING.CLOSE.CONNECTIONS DEV)
                  (fetch (FDEV DEVICENAME) of DEV)))
       (T (LET [(DEV (OR DEVICE (\GETDEVICEFROMNAME (\CANONICAL.NSHOSTNAME HOST)
                                    T T]
             (COND
                (DEV (\NSFILING.CLOSE.CONNECTIONS DEV)
                     T])

)

(ADDTOVAR \AFTERLOGINFNS \NSFILING.AFTER.LOGIN)
```

;; Support

```
(DEFINEQ
```

(\NSFILING.CONNECT

```
  [LAMBDA (SESSION DIRPATH REALREQUIRED CREATE?)                    ; Edited 14-Sep-87 14:06 by bvm:
```

;; Follow the list of directories in DIRPATH and return the handle for the final one. The special case when DIRPATH is NIL is equivalent to
;; connecting to the root directory. Uses cached paths to avoid useless reconnecting.

```
    (PROG (NEW.HANDLE NSPATHNAME)
          [COND
             ((SETQ NEW.HANDLE (\NSFILING.LOOKUP.CACHE SESSION DIRPATH))
                                                                    ; Nothing needs to be done because we're already connected to
                                                                    ; this path.
              (RETURN (AND NEW.HANDLE (fetch NSHDIRECTORYP of NEW.HANDLE)
                           NEW.HANDLE]
          [SETQ NSPATHNAME (COND
                              [(CDR DIRPATH)
                               (CONCATLIST (CDR (for DIR in DIRPATH join (LIST '/ DIR]
                              (T (CAR DIRPATH]
          (SETQ NEW.HANDLE (FILING.CALL SESSION 'OPEN [AND NSPATHNAME `((PATHNAME ,NSPATHNAME]
                                \NSFILING.NULL.HANDLE NIL SESSION 'RETURNERRORS))
          (SELECTQ (CAR NEW.HANDLE)
               (NIL                                                 ; Utter failure
                    (RETURN))
               (ERROR (COND
                         ((AND (EQ (CADDR NEW.HANDLE)
                                   'FileNotFound)
                               (SETQ NEW.HANDLE (\NSFILING.MAYBE.CREATE SESSION DIRPATH CREATE?)))
                                                                    ; Successfully created

                          )
                         (T                                         ; Failed for some other reason
                            (RETURN))))
               NIL)
          (RETURN (AND [NLISTP (SETQ NEW.HANDLE (\NSFILING.ADD.TO.CACHE SESSION
                                                  (create FILINGHANDLE
                                                          NSHDIRECTORYPATH _ DIRPATH
                                                          NSHDATUM _ NEW.HANDLE]
                       (fetch NSHDIRECTORYP of NEW.HANDLE)
                       NEW.HANDLE])
```

(\NSFILING.MAYBE.CREATE

```
  [LAMBDA (SESSION DIRLST CREATE?)                                  ; Edited  1-Jun-87 16:06 by bvm:
```

;;; Called to possibly create a nonexistent subdirectory. DIRLST is a list of subdirectories from root to leaf.

```
    (LET (OLDHANDLE NEW.DIR)
         (AND (SELECTQ CREATE?
                   (:ASK (SETQ CREATE? :ASKED)                      ; flag needed on recursive calls to show we asked up here
                         (EQ 'Y (ASKUSER DWIMWAIT 'Y (CONCAT "Create subdirectory {" (fetch FSNAMESTRING
                                                                                       of SESSION)
                                                             "}<"
                                                             [CONCATLIST (for DIR in DIRLST
                                                                            join (LIST DIR '>]
                                                             "? "))))
                   (NIL NIL)
                   T)
              (SETQ OLDHANDLE (\NSFILING.CONNECT SESSION (for TAIL on DIRLST collect (CAR TAIL)
                                                            while (CDR TAIL) finally (SETQ NEW.DIR (CAR TAIL)))
                                     T CREATE?))
              (COND
                 ((AND (SETQ OLDHANDLE (FILING.CALL SESSION 'CREATE (fetch NSHDATUM of OLDHANDLE)
                                                    `((NAME ,(\NSFILING.REMOVEQUOTES NEW.DIR))
                                                      (IS.DIRECTORY T)
                                                      (FILE.TYPE 1))
                                                    NIL SESSION 'RETURNERRORS))
                       (NEQ (CAR OLDHANDLE)
                            'ERROR))                                ; Success
                  OLDHANDLE)
                 (T (SELECTQ CREATE?
                         ((:ASK :ASKED)                             ; Interactive use--let user know why we failed.
                              (CL:ERROR "Could not create ~A because of ~A: ~A"
```

```
                                    [CONCATLIST (LIST* "{" (fetch FSNAMESTRING of SESSION)
                                                       "}<"
                                                       (for DIR in DIRLST join (LIST DIR '>]
                                    (CADR OLDHANDLE)
                                    (STRING (CADDR OLDHANDLE)))
                               NIL)
                          NIL])
```

(\NSFILING.REMOVEQUOTES

```
[LAMBDA (NAME)                                                    (* bvm%: "24-Sep-85 15:17")
```

;;; Removes quoting characters from NAME

```
    (COND
       [(STRPOS "'" NAME)
        (CONCATCODES (bind (I _ 0)
                           CH while (SETQ CH (NTHCHARCODE NAME (add I 1)))
                     collect (COND
                                ((EQ CH (CHARCODE %'))
                                 (OR (NTHCHARCODE NAME (add I 1))
                                     CH))
                                (T CH]
       (T NAME])
```

(\NSFILING.ADDQUOTES

```
[LAMBDA (NAME ALREADYQUOTED)                                      (* bvm%: "27-Jun-86 11:16")
```

;;; Returns NAME with funny characters (file name delimeters) quoted. If ALREADYQUOTED is true, then any quote characters in NAME are interpreted
;;; as quoting the next char, rather than being a funny char that needs to be quoted

```
    (COND
       [[for CH instring (OR (STRINGP NAME)
                             (SETQ NAME (MKSTRING NAME)))
           bind QUOTED do (COND
                             (QUOTED (SETQ QUOTED (SETQ CH NIL)))
                             (T (SELCHARQ CH
                                     ((%: ; < > } %] /)
                                      (RETURN T))
                                     (%' (COND
                                            (ALREADYQUOTED (SETQ QUOTED T))
                                            (T (RETURN T))))
                                     NIL)))
           finally (COND
                      ((EQ CH (CHARCODE "."))
                       ;; Name ending in period, the period is significant and must be quoted, else we would leave it out as being an extensionless
                       ;; file indicator
                       (RETURN T]                                 ; Yes, there is something funny, so it's worth constructing a
                                                                  ; whole new name
        (CONCATCODES (for CH instring NAME bind QUOTED NAMECHARS LASTCH
                        do (COND
                              (QUOTED (SETQ QUOTED NIL))
                              (T (SELCHARQ (SETQ LASTCH CH)
                                      ((%: ; < > } %] /)
                                       (push NAMECHARS (CHARCODE %')))
                                      (%' [COND
                                             (ALREADYQUOTED (SETQ QUOTED T))
                                             (T (push NAMECHARS (CHARCODE %']
                                     NIL)))
                           (push NAMECHARS CH)
                        finally [COND
                                   ((EQ LASTCH (CHARCODE "."))     ; See ugliness above
                                    (RPLACD NAMECHARS (CONS (CHARCODE %')
                                                            (CDR NAMECHARS]
                                (RETURN (REVERSE NAMECHARS]
       (T NAME])
```

(\FILING.ATTRIBUTE.TYPE.SEQUENCE

```
[LAMBDA (ATTRIBUTETYPES)                                          (* ecc " 3-AUG-83 16:39")
   (for ATTR in ATTRIBUTETYPES collect (\FILING.ATTRIBUTE.TYPE ATTR])
```

(\FILING.ATTRIBUTE.TYPE

```
[LAMBDA (ATTR NOERRORFLG)                                         ; Edited  3-Jun-87 16:34 by bvm:
   (OR (FIXP ATTR)
       (for X in \NSFILING.ATTRIBUTES do [COND
                                            ((EQ (CAR X)
                                                 ATTR)
                                             (RETURN (CADR X]
          finally (OR NOERRORFLG (ERROR "Unknown Filing attribute" ATTR])
```

(\LISP.TO.NSFILING.ATTRIBUTE

```
[LAMBDA (ATTRIBUTE VALUE)                                         ; Edited 18-Apr-88 15:00 by bvm
```

;; Convert a Lisp file attribute and its value to a filing attr/val pair (list of two elements), or NIL if we can't figure it out.

```
(LET
 (X)
 [SELECTQ ATTRIBUTE
      (WRITEDATE (SETQ ATTRIBUTE 'MODIFIED.ON)
                 (SETQ VALUE (OR (IDATE VALUE)
                                 (LISPERROR "ILLEGAL ARG" VALUE))))
      (READDATE (SETQ ATTRIBUTE 'READ.ON)
                (SETQ VALUE (OR (IDATE VALUE)
                                (LISPERROR "ILLEGAL ARG" VALUE))))
      (CREATIONDATE (SETQ ATTRIBUTE 'CREATED.ON)
                    (SETQ VALUE (OR (IDATE VALUE)
                                    (LISPERROR "ILLEGAL ARG" VALUE))))
      (TYPE (SETQ ATTRIBUTE 'FILE.TYPE)
            (SETQ VALUE (OR (\FILETYPE.FROM.TYPE VALUE)
                            \NSFILING.TYPE.BINARY)))
      (SIZE (SETQ ATTRIBUTE 'SIZE.IN.BYTES)
            (SETQ VALUE (UNFOLD VALUE BYTESPERPAGE)))
      (COND
         ((SETQ X (ASSOC ATTRIBUTE \LISP.TO.NSFILING.ATTRIBUTES))
          (SETQ ATTRIBUTE (CADR X]
 (COND
    ((SETQ X (CADDR (ASSOC ATTRIBUTE \NSFILING.ATTRIBUTES))) ; Check that the value is reasonable before we die deep in a
                                                             ; courier call.  X = courier type.
     (LET
      ((COERCEDVALUE VALUE))
      (OR (SELECTQ X
                ((CARDINAL UNSPECIFIED)
                     (AND (SMALLP VALUE)
                          (>= VALUE 0)))
                (INTEGER (SMALLP VALUE))
                ((LONGCARDINAL LONGINTEGER TIME)
                     (FIXP VALUE))
                (USER (SETQ COERCEDVALUE (PARSE.NSNAME VALUE)))
                (ACCESS.LIST                                 ; A protection value is complicated
                            [AND (OR (NULL VALUE)
                                     (LISTP VALUE))
                                 (SETQ COERCEDVALUE
                                  (for ENTRY in VALUE
                                     collect [LIST (PARSE.NSNAME (CAR ENTRY))
                                                   (COND
                                                      ([AND (LISTP (CADR ENTRY))
                                                            (for ACCESS in (CADR ENTRY)
                                                               always (MEMB ACCESS
                                                                            '(ALL READ WRITE OWNER ADD REMOVE]
                                                       (CADR ENTRY))
                                                      (T (RETURN]
                                     finally (RETURN (COURIER.CREATE (FILING . ACCESS.LIST)
                                                                     ENTRIES _ $$VAL DEFAULTED _ NIL])
                (PROGN                                       ; accept anything, hope for the best
                       T))
          (\ILLEGAL.ARG VALUE))
      (LIST ATTRIBUTE COERCEDVALUE)))
    ((FIXP ATTRIBUTE)                                        ; This is how to get raw, unregistered attributes.  Value must be
                                                             ; sequence unspecified
     (COND
        ([for (TL _ VALUE) by (CDR TL) while TL always (AND (LISTP TL)
                                                            (SMALLP (CAR TL]
         (LIST ATTRIBUTE VALUE))
        (T (\ILLEGAL.ARG VALUE])

)

;; FILINGHANDLE stuff

(DEFINEQ

(\**NSFILING.GETFILE**
  [LAMBDA (DEVICE FILENAME ACCESS RECOG OPTION PARAMETERS DIROK SEQUENTIAL OLDSTREAM)
                                                             ; Edited 23-May-2024 23:12 by frank
                                                             ; Edited 19-Aug-88 17:17 by bvm

    ;; Opens FILENAME for specified ACCESS and RECOG, returning a stream.  If OPTION is NAME, ATTRIBUTES, or HANDLE, just return the
    ;; appropriate information instead of a stream.  If OPTION is DIRECTORY, return T or NIL if FILENAME is a directory or not -- PARAMETERS
    ;; gives the CREATE? option in case the directory doesn't exist.  If ACCESS is not NONE, then PARAMETERS gives extra parameters for the
    ;; open.

    (RESETLST
        [PROG ((NAME "")
               VERSION EXPLICIT-VERSION SESSION OLDHANDLE FILE.ID HANDLE FILESTREAM FULLNAME PARSE ROOTNAME
               DIRPATH RANDEVICE HAVELOCK SERIALIZE)
              (COND
                 ((NOT (SETQ SESSION (\**GETFILINGCONNECTION** DEVICE)))
                  (RETURN NIL)))
              [COND
                 ((EQ ACCESS 'SERIALIZE)                     ; Like INPUT, but retrieve a serialized stream on file
                  (SETQ ACCESS 'INPUT)
```

```
                (SETQ SERIALIZE 'SERIALIZE)
                (SETQ SEQUENTIAL T))
               ((AND (NOT SEQUENTIAL)
                     (NOT OPTION)
                     *NSFILING-RANDOM-ACCESS*)                ; RANDEVICE set if we want to open a randaccess stream
                (SETQ RANDEVICE (fetch NSRANDOMDEVICE of (fetch DEVICEINFO of DEVICE]
          RETRY
              [COND
                 [(SETQ HANDLE (\NSFILING.LOOKUP.CACHE SESSION FILENAME))
                                                              ; Cache hit
                  (COND
                     (OPTION                                  ; Got handle, so just do what the option said (else fall thru and try
                                                              ; to open a file)
                             (GO HANDLE.OPTION]
                 ((AND (LISTP FILENAME)
                       (EQ (CAR FILENAME)
                           'FILE.ID))                         ; Identifying file by ID, take shortcut.  Do this second just in case
                                                              ; we have cached this file already
                  (SETQ FILE.ID (CADR FILENAME)))
                 (T                                           ; Parse the name and go thru all this hassle
                    (SETQ PARSE (\NSFILING.PARSE.FILENAME FILENAME))
                    (SETQ DIRPATH (fetch NSDIRECTORIES of PARSE))
                    (COND
                       ((NULL DIRPATH)                        ; No directories specified, so is illegal name
                        (GO FILE.NOT.FOUND))
                       [(EQ OPTION 'DIRECTORY)
                        (RETURN (AND (fetch NSDIRECTORYP of PARSE)
                                     (SETQ HANDLE (\NSFILING.CONNECT SESSION DIRPATH T PARAMETERS))
                                     (GO HANDLE.OPTION]
                       ((AND (fetch NSDIRECTORYP of PARSE)
                             (NOT DIROK))                     ; No name, just a directory.  Failure unless caller said a directory
                                                              ; file is ok
                        (GO FILE.NOT.FOUND)))
                    (SETQ EXPLICIT-VERSION (fetch NSVERSION of PARSE))
                    (SETQ ROOTNAME (fetch NSROOTNAME of PARSE]
              [COND
                 (HANDLE                                      ; We have an open file handle from the cache
                         )
                 [FILE.ID                                     ; Try to open an existing file by ID.
                         (COND
                            ([SETQ HANDLE (\NSFILING.OPEN.HANDLE SESSION `((FILE.ID ,FILE.ID))
                                                 (AND RANDEVICE (SELECTQ ACCESS
                                                                     ((BOTH APPEND)
                                                                         'OUTPUT)
                                                                     ACCESS]
                             (SETQ HAVELOCK RANDEVICE))
                            (T (GO FILE.NOT.FOUND]
                 (T                                           ; open by name
                    (SETQ OLDHANDLE (\NSFILING.OPEN.HANDLE SESSION [\NSFILING.COMPOSE.PATHNAME
                                                                        DIRPATH ROOTNAME
                                                                        (OR EXPLICIT-VERSION
                                                                            (SELECTQ RECOG
                                                                                (OLDEST '-)
                                                                                '+]
                                                 (AND RANDEVICE (SETQ HAVELOCK
                                                                      (SELECTQ ACCESS
                                                                          ((OUTPUT BOTH APPEND)
                                                                           ; When opening for output, only get lock right now if we know we
                                                                           ; will be playing with the old file.
                                                                              (AND (OR EXPLICIT-VERSION
                                                                                       (NEQ RECOG 'NEW))
                                                                                   'OUTPUT))
                                                                          (INPUT ACCESS)
                                                                          NIL)))
                                                 'RETURNERRORS))
                       (COND
                          [[OR (NULL OLDHANDLE)
                               (AND (LISTP OLDHANDLE)
                                    (EQ (CADR OLDHANDLE)
                                        'ACCESS.ERROR)
                                    (EQ (CADDR OLDHANDLE)
                                        'FileNotFound]          ; No file of any version exists by this name
                           (SETQ HAVELOCK NIL)
                           (SELECTQ RECOG
                                ((OLD OLDEST)                 ; No version exists, so certainly this one doesn't
                                     (RETURN NIL))
                                (COND
                                   ((EQ ACCESS 'INPUT)        ; Version given explicitly, file does not exist
                                    (RETURN NIL))
                                   ((NULL EXPLICIT-VERSION)   ; No extant version, so create number 1
                                    (OR RANDEVICE (SETQ VERSION 1)))
                                   (T (SETQ VERSION EXPLICIT-VERSION]
                          ((LISTP OLDHANDLE)                  ; Error case
                           (SETQ HAVELOCK NIL)
                           (SETQ FILESTREAM OLDHANDLE)
                           (GO HANDLE.ERROR))
                          ((AND (fetch NSHDIRECTORYP of OLDHANDLE)
```

```
                        (NOT DIROK))                          ; It's a directory, don't try to treat as ordinary file
                    (GO FILE.NOT.FOUND))
                   [(OR EXPLICIT-VERSION (NEQ RECOG 'NEW'))   ; Old file exists, use it unless we explicitly requested a new
                                                              ; version
                    (SETQ HANDLE OLDHANDLE)
                    (COND
                       (EXPLICIT-VERSION (SETQ VERSION EXPLICIT-VERSION]
                   ((NOT RANDEVICE)                           ; RECOG = NEW -- write a file version one higher.  Don't do this
                                                              ; in random access case, because we can get the server to tell
                                                              ; us the version there.
                    (SETQ VERSION (ADD1 (OR [CADR (ASSOC 'VERSION (OR (fetch NSHATTRIBUTES of OLDHANDLE)
                                                                      (\NSFILING.FILLIN.ATTRIBUTES SESSION
                                                                            OLDHANDLE]
                                            (GO FILE.NOT.FOUND]
;; At this point, HANDLE is an open handle on the file we want, or is NIL in the case where we are to create a new version, in which case
;; VERSION has been set correctly.
          (SETQ FULLNAME (\NSFILING.FULLNAME SESSION (OR HANDLE PARSE)
                                 VERSION))
          (COND
             (OPTION                                          ; Not opening file, something simpler
                    (GO HANDLE.OPTION))
             ((AND HANDLE (NOT OLDSTREAM)
                   (\NSFILING.CONFLICTP DEVICE SESSION HANDLE ACCESS))
              (GO FILE.BUSY)))
          (SELECTQ ACCESS
              (INPUT (COND
                         ((NULL HANDLE)                       ; Odd to get here.  E.g., open for INPUT recog NEW.
                          (GO FILE.NOT.FOUND))
                         (RANDEVICE (SETQ FILESTREAM (\NSRANDOM.CREATE.STREAM SESSION HANDLE RANDEVICE
                                                              'INPUT HAVELOCK OLDSTREAM)))
                         [(NEQ (fetch NSHACCESS of HANDLE)
                               'OUTPUT)                       ; Just retrieve old file
                          (SETQ FILESTREAM (FILING.CALL SESSION (OR SERIALIZE 'RETRIEVE)
                                                        (fetch NSHDATUM of HANDLE)
                                                        NIL SESSION 'RETURNERRORS 'KEEPSTREAM]
                         (T (GO FILE.BUSY))))
              ((OUTPUT BOTH APPEND)
                   (COND
                      ((AND (NEQ ACCESS 'OUTPUT)
                            (NOT RANDEVICE))                  ; Sequential can only write whole files
                       (GO FILE.WONT.OPEN)))
                   (COND
                      [HANDLE                                 ; File already exists, need to overwrite
                              (COND
                                 (RANDEVICE (SETQ FILESTREAM (\NSRANDOM.CREATE.STREAM SESSION HANDLE
                                                                    RANDEVICE ACCESS HAVELOCK OLDSTREAM T)))
                                 [(NULL (fetch NSHACCESS of HANDLE))
                                                              ; Overwrite existing file sequentially
                                  [SETQ FILESTREAM (OR (\NSFILING.CHECK.ACCESS SESSION HANDLE 'WRITE)
                                                       (FILING.CALL SESSION 'REPLACE (fetch NSHDATUM
                                                                                          of HANDLE)
                                                                    NIL NIL SESSION 'RETURNERRORS 'KEEPSTREAM]
                                  (COND
                                     ((type? STREAM FILESTREAM)
                                                              ; Cache of saved attributes is now wrong
                                      (replace NSHATTRIBUTES of HANDLE with NIL)
                                                              ; Save attributes to change after file is stored
                                      (replace NSFILING.NEW.ATTRIBUTES of FILESTREAM with PARAMETERS]
                                 (T (GO FILE.BUSY]
                      (OLDSTREAM                              ; Trying to reopen old stream, failed.
                              (RETURN NIL))
                      [(SETQ OLDHANDLE (\NSFILING.CONNECT SESSION DIRPATH T T))
                                                              ; Need to create the file, so first had to get a handle on the
                                                              ; parent (CREATE and STORE procedures do not permit
                                                              ; PATHNAME as one of the specifying attributes).
                       (COND
                          [RANDEVICE                          ; Create a new, empty file, then start writing pages to it.  Lucky
                                                              ; us, we can do this right for a change.
                                  (SETQ HANDLE (FILING.CALL
                                                   SESSION
                                                   'CREATE
                                                   (fetch NSHDATUM of OLDHANDLE)
                                                   `([NAME ,(\NSFILING.REMOVEQUOTES (fetch NSROOTNAME
                                                                                       of PARSE]
                                                     ,@[AND VERSION `((VERSION ,VERSION]
                                                     ,@PARAMETERS)
                                                   '((LOCK EXCLUSIVE))
                                                   SESSION
                                                   'RETURNERRORS))
                                  (COND
                                     ([OR (NLISTP HANDLE)
                                          (EQ (CAR HANDLE)
                                              'ERROR)
                                          (LISTP (SETQ HANDLE (\NSFILING.ADD.TO.CACHE SESSION
                                                                    (create FILINGHANDLE
                                                                            NSHDATUM _ HANDLE
```

```
                                                                    NSHACCESS _ 'OUTPUT]
                                                                ; Create failed or we can't read its attributes!  Fall thru to error
                                                                ; handler
                                        (SETQ FILESTREAM HANDLE)
                                        (GO HANDLE.ERROR))
                                       ((type? STREAM (SETQ FILESTREAM (\NSRANDOM.CREATE.STREAM SESSION
                                                                                    HANDLE RANDEVICE ACCESS T)))
                                                                ; Succeeded in opening stream, i.e., no further conflicts detected.
                                        (SETQ FULLNAME (\NSFILING.FULLNAME SESSION HANDLE)))
                                       (T (GO HANDLE.ERROR]
                                (T                              ; Start writing new file, guessing the version.  Ideally we shouldn't
                                                                ; guess the version, but Lisp wants a full file name NOW
                                                                ; (grumble).
                                   (SETQ FILESTREAM
                                    (OR (\NSFILING.CHECK.ACCESS SESSION OLDHANDLE 'ADD)
                                        (FILING.CALL SESSION 'STORE (fetch NSHDATUM of OLDHANDLE)
                                               `([NAME ,(\NSFILING.REMOVEQUOTES (fetch NSROOTNAME of PARSE]
                                                 (VERSION ,VERSION)
                                                 ,@PARAMETERS)
                                               NIL NIL SESSION 'RETURNERRORS 'KEEPSTREAM]
                           (T (GO FILE.NOT.FOUND))))
                     (\ILLEGAL.ARG ACCESS))
          (COND
             ((NOT (type? STREAM FILESTREAM))                   ; Had handle, but failed to open it.
              (GO HANDLE.ERROR)))
          (replace FULLFILENAME of FILESTREAM with (COND
                                                     (*UPPER-CASE-FILE-NAMES* (MKATOM (U-CASE FULLNAME)))
                                                     (T FULLNAME)))
          (replace NSFILING.CONNECTION of FILESTREAM with SESSION)
          (replace NSFILING.HANDLE of FILESTREAM with HANDLE)
          (replace DEVICE of FILESTREAM with (OR RANDEVICE DEVICE))
          (COND
             (HANDLE (add (fetch NSHBUSYCOUNT of HANDLE)
                          1)))
          (RETURN FILESTREAM)
    HANDLE.OPTION

;; Come here when we have obtained the handle on the file in question, but OPTION is non-NIL, so we want to do something other than
;; open a file.

          (RETURN (SELECTQ OPTION
                       (NAME (if HANDLE
                                 then (\NSFILING.FULLNAME SESSION HANDLE NIL *UPPER-CASE-FILE-NAMES*)
                                 else                           ; OUTFILEP case: no handle, but we have computed the name
                                      FULLNAME))
                       (DIRECTORY                               ; I'm pretty sure HANDLE can't be NIL at this point, but a little
                                                                ; test never hurt anyone.
                                  (AND HANDLE (fetch NSHDIRECTORYP of HANDLE)
                                       (\NSFILING.FULLNAME SESSION HANDLE NIL *UPPER-CASE-FILE-NAMES*)))
                       (ATTRIBUTES (OR (fetch NSHATTRIBUTES of HANDLE)
                                       (\NSFILING.FILLIN.ATTRIBUTES SESSION HANDLE)))
                       (HANDLE (CL:FUNCALL PARAMETERS SESSION HANDLE))
                       (SHOULDNT)))
    HANDLE.ERROR

;; Come here with FILESTREAM set to an error returned from some courier call

          (COND
             ([NOT (EQUAL FILESTREAM '(ERROR SESSION.ERROR TokenInvalid]
              (COND
                 (HAVELOCK (\NSRANDOM.RELEASE.LOCK SESSION HANDLE)))
              (RETURN (\NSFILING.HANDLE.ERROR SESSION FILESTREAM FILENAME)))
             ((SETQ SESSION (\NSFILING.GET.NEW.SESSION SESSION DEVICE))
                                                                ; Got new session, so start over.  Note that we may have to
                                                                ; reparse, since the first time thru we might have gotten the
                                                                ; cached handle.
              (SETQ HAVELOCK (SETQ HANDLE (SETQ VERSION NIL)))
              (GO RETRY))
             (T                                                 ; Can't get connection at all?  OH well, die as if it were true from
                                                                ; the start.
                (RETURN NIL)))
    FILE.NOT.FOUND
          (COND
             (HAVELOCK (\NSRANDOM.RELEASE.LOCK SESSION HANDLE)))
          (RETURN NIL)
    FILE.BUSY
          (COND
             (HAVELOCK (\NSRANDOM.RELEASE.LOCK SESSION HANDLE)))
    FILE.WONT.OPEN
          (RETURN (WITHOUT.SESSION.MONITOR SESSION (LISPERROR "FILE WON'T OPEN" FULLNAME])])
```

(\NSFILING.LOOKUP.CACHE

```
  [LAMBDA (CONNECTION FILENAME)                                 ; Edited  9-Jun-87 22:55 by bvm:
    (LET ((CACHE (fetch FSCACHEDHANDLES of CONNECTION))
          ENTRY)
         (COND
```

```
            ([COND
                ((EQ (CAR (LISTP FILENAME))
                     'FILE.ID)                                    ; Look by id
                 (find old ENTRY in CACHE bind (ID _ (CADR FILENAME)) suchthat (EQUAL (fetch NSHFILEID of ENTRY)
                                                                                      ID)))
                [(OR (NULL FILENAME)
                     (LISTP FILENAME))                            ; Looking for directory match
                 (find old ENTRY in CACHE bind NAME (PATHLENGTH _ (LENGTH FILENAME))
                    suchthat (AND (NEQ (SETQ NAME (fetch NSHDIRECTORYPATH of ENTRY))
                                       T)
                                  (EQ (LENGTH NAME)
                                      PATHLENGTH)
                                  (for X in FILENAME always (STRING-EQUAL X (pop NAME]
                (T                                                ; Looking for file name match
                   (find old ENTRY in CACHE suchthat (STRING-EQUAL (fetch NSHNAME of ENTRY)
                                                                   FILENAME]
             [COND
                ((CDR CACHE)                                      ; Promote to front of cache
                 (replace FSCACHEDHANDLES of CONNECTION with (CONS ENTRY (DREMOVE ENTRY CACHE]
             ENTRY])
```

(\NSFILING.ADD.TO.CACHE

```
  [LAMBDA (SESSION HANDLE NOERRORFLG)                             ; Edited 1-Sep-87 11:42 by bvm:
```

;;; Add file HANDLE to the cache for SESSION and return it, or an earlier cached version of the same handle if there is one

```
    (PROG ((CACHE (fetch FSCACHEDHANDLES of SESSION))
           (ID (fetch NSHFILEID of HANDLE))
           OLDHANDLE)
          [COND
             ((NULL ID)
              (COND
                 ((OR (NLISTP (SETQ ID (\NSFILING.FILLIN.ATTRIBUTES SESSION HANDLE NOERRORFLG)))
                      (EQ (CAR ID)
                          'ERROR))                                ; Pass error up
                  (RETURN ID)))
              (SETQ ID (fetch NSHFILEID of HANDLE]
          (COND
             ([AND ID (SETQ OLDHANDLE (find H in CACHE suchthat (EQUAL (fetch NSHFILEID of H)
                                                                       ID]
                                                                  ; Don't keep duplicates--flush the new one and return the old one
              (\NSFILING.CLOSE.HANDLE SESSION HANDLE)
              (RETURN OLDHANDLE)))
          [while (GREATERP (LENGTH CACHE)
                           FILING.CACHE.LIMIT)
             do                                                   ; Flush old unused handles to keep the cache within limits
                (for H in CACHE when (EQ (fetch NSHBUSYCOUNT of H)
                                         0)
                   do (SETQ OLDHANDLE H))
                (COND
                   (OLDHANDLE                                   ; The least recently referenced unused handle
                          (SETQ CACHE (DREMOVE OLDHANDLE CACHE))
                          (\NSFILING.CLOSE.HANDLE SESSION OLDHANDLE)
                          (SETQ OLDHANDLE NIL))
                   (T                                           ; All handles are busy
                      (RETURN]
          (replace FSCACHEDHANDLES of SESSION with (CONS HANDLE CACHE))
          (RETURN HANDLE])
```

(\NSFILING.OPEN.HANDLE

```
  [LAMBDA (SESSION PNAME.OR.PROPS CONTROLS NOERRORFLG PARENT)     ; Edited 19-Aug-88 17:38 by bvm
    (LET [(HANDLE (FILING.CALL SESSION 'OPEN [OR (LISTP PNAME.OR.PROPS)
                                                 `((PATHNAME ,PNAME.OR.PROPS]
                               (if PARENT
                                   then (fetch NSHDATUM of PARENT)
                                 else \NSFILING.NULL.HANDLE)
                               [AND CONTROLS `((LOCK ,(SELECTQ CONTROLS
                                                               (INPUT 'SHARE)
                                                               (OUTPUT 'EXCLUSIVE)
                                                               (SHOULDNT]
                               SESSION
                               (OR NOERRORFLG 'NOERROR]
          (COND
             ((OR (NLISTP HANDLE)
                  (EQ (CAR HANDLE)
                      'ERROR))                                    ; Failure return
              HANDLE)
             (T (LET ((RESULT (\NSFILING.ADD.TO.CACHE SESSION (SETQ HANDLE
                                                                    (create FILINGHANDLE
                                                                            NSHDATUM _ HANDLE
                                                                            NSHACCESS _ CONTROLS))
                                                      NOERRORFLG)))
                   [COND
                      ((NOT (TYPENAMEP RESULT 'FILINGHANDLE))     ; Error trying to get attributes--close the handle altogether now,
                                                                  ; since it's not going into the cache.
```

```
                        (\NSFILING.CLOSE.HANDLE SESSION HANDLE))
                      (CONTROLS                                    ; May need to release lock if there's an error later.
                             (RESETSAVE NIL (LIST (FUNCTION \NSRANDOM.RELEASE.IF.ERROR)
                                                  SESSION HANDLE]
                    RESULT])
```

(\NSFILING.CONFLICTP

```
  [LAMBDA (DEVICE SESSION HANDLE ACCESS)                           ; Edited 19-Aug-88 17:17 by bvm
```

;; True if opening HANDLE on SESSION for specified ACCESS would present an access conflict for streams already open on DEVICE. We need ;; this as an explicit check because we might have files open on expired sessions where we haven't yet reestablished their streams on the new ;; session, and hence the handle conflict would not be apparent.

```
    (LET ((OPENFILES (fetch (FDEV OPENFILELST) of DEVICE)))
         (AND OPENFILES (for S in OPENFILES bind (NAME _ (\NSFILING.FULLNAME SESSION HANDLE))
                           when (STRING-EQUAL NAME (fetch FULLFILENAME of S))
                           do                                      ; Note that looking at one stream on the file is sufficient for
                                                                   ; conflict check.
                              (RETURN (SELECTQ ACCESS
                                               ((OUTPUT BOTH APPEND)
                                                                   ; Always a conflict
                                                   T)
                                               (INPUT              ; Ok if only input
                                                      (DIRTYABLE S))
                                               (\ILLEGAL.ARG ACCESS])
```

(\NSFILING.CHECK.ACCESS

```
  [LAMBDA (SESSION HANDLE TYPE)                                    ; Edited 30-Nov-87 10:39 by bvm:
```

;; Check that user has TYPE access to the specified file handle. TYPE is one of the values of the ACCESS control: READ, WRITE, OWNER, ADD, ;; REMOVE, ALL. If user has access, returns NIL; otherwise, returns some sort of courier error.

;; In Filing 4 (Services 8.0) this can't work, so we pretend it succeeds.

```
    (AND (NEQ (fetch FSPROTOCOLNAME of SESSION)
              'FILING.4)
         (LET [(RESULT (FILING.CALL SESSION 'GET.CONTROLS (fetch NSHDATUM of HANDLE)
                                    '(ACCESS)
                                    SESSION
                                    'RETURNERRORS]
              (COND
                 ((EQ (CAR RESULT)
                      'ERROR)
                  RESULT)
                 ([NOT (for A in (CADR (ASSOC 'ACCESS RESULT)) thereis (OR (EQ A TYPE)
                                                                           (EQ A 'ALL]
                                                                   ;; Fake a protection error.  Don't generate the error here, because caller may need to release a lock first.  The ASSOC is
                                                                   ;; because filing returns a list of controls, even though I only asked for one (bug).
                  '(ERROR ACCESS.ERROR AccessRightsInsufficient])
```

(\NSFILING.FILLIN.ATTRIBUTES

```
  [LAMBDA (SESSION HANDLE NOERRORFLG)                              ; Edited  3-Jun-87 19:25 by bvm:
    (OR (fetch NSHATTRIBUTES of HANDLE)
        (LET [(ATTRS (FILING.CALL SESSION 'GET.ATTRIBUTES (fetch NSHDATUM of HANDLE)
                                  \NSFILING.USEFUL.ATTRIBUTE.TYPES SESSION (OR NOERRORFLG 'RETURNERRORS]
             (COND
                ((AND (LISTP ATTRS)
                      (NEQ (CAR ATTRS)
                           'ERROR))
                 (replace NSHATTRIBUTES of HANDLE with ATTRS)
                 (for X in ATTRS do                                ;; Fill in interesting attributes that we might want to get at quickly and not lose if a SETFILEINFO is done
                                    (SELECTQ (CAR X)
                                         (PATHNAME (replace NSHPATHNAME of HANDLE with (CADR X)))
                                         (FILE.ID (replace NSHFILEID of HANDLE with (CADR X)))
                                         (IS.DIRECTORY (replace NSHDIRECTORYP of HANDLE with (CADR X)))
                                         NIL)))
                ((NOT NOERRORFLG)
                 (COURIER.SIGNAL.ERROR (fetch FSPROTOCOLNAME of SESSION)
                        'GET.ATTRIBUTES ATTRS)))
             ATTRS])
```

(\NSFILING.COMPOSE.PATHNAME

```
  [LAMBDA (DIRPATH NAME VERSION)                                   (* bvm%: "19-Dec-85 16:55")
```

;;; Makes a NS pathname out of the file name with given components. All components are assumed to be quoted as needed. NAME and/or VERSION ;;; can be NIL

```
    (CONCATLIST (NCONC (CDR (for DIR in DIRPATH join (LIST '/ DIR)))
                       (AND NAME (LIST '/ (\NSFILING.ADDQUOTES NAME T)))
                       (AND VERSION (LIST '! VERSION])
```

(\NSFILING.PARSE.FILENAME

```
  [LAMBDA (FILENAME PATTERNP)                                      ; Edited 10-Dec-87 11:09 by bvm:
```

;;; Parses FILENAME into an NSFILINGPARSE record. Hate to do this independent of UNPACKFILENAME, but there's too much to worry about -- need ;;; to parse the directories individually, require periods not to mean version, ignore colon as a device delimeter, etc.

;;; PATTERNP is true when parsing a directory pattern. Main difference is we preserve final dot in name so caller knows it has to be extensionless.

;;; Returns NIL if filename is bad.

```
(bind CH (I _ 1)
      (NC _ (NCHARS FILENAME))
      START VERSION SEMI DOTSEEN DIRS END LASTHOST NAME
   first (COND
           ([OR (NULL (SETQ LASTHOST (SELCHARQ (CHCON1 FILENAME)
                                               ({ (CHARCODE }))
                                               (%( (CHARCODE %)))
                                               (%[ (CHARCODE %]))
                                               NIL)))
                (until (EQ (SETQ CH (NTHCHARCODE FILENAME (add I 1)))
                           LASTHOST)
                   do (COND
                         ((NULL CH)                          ; end of file name
                          (RETURN T]                         ; Bad file name
            (RETURN NIL)))
        [SETQ START (+ I (SELCHARQ (NTHCHARCODE FILENAME (ADD1 I))
                                   ((/ <)
                                        2)
                                   (PROGN                    ; No directory
                                          1]
   while (<= (add I 1)
             NC)
   do (SELCHARQ (NTHCHARCODE FILENAME I)
            (;                                               ; Version marker maybe
               (SETQ SEMI I))
            (%. (OR DOTSEEN (SETQ DOTSEEN I)))
            (%'                                              ; quote mark, skip it and next char
                (add I 1))
            ((/ >)                                           ; Directory marker
                 (if SEMI
                     then                                    ; Version marker inside directory?
                         (RETURN NIL))
                 [SETQ DIRS (NCONC1 DIRS (SUBSTRING FILENAME START (SUB1 I]
                 (SETQ SEMI (SETQ DOTSEEN NIL))
                 (SETQ START (ADD1 I)))
            (* (if (NOT PATTERNP)
                   then (RETURN NIL)))
           NIL)
   finally [COND
             ((NEQ START I)
              [SETQ END (COND
                           (SEMI (SUB1 SEMI))
                           (T (SUB1 I]
              [COND
                 ((AND (EQ END DOTSEEN)
                       (NOT PATTERNP))                       ; Don't include final dot of extensionless files in actual name on
                                                             ; server
                  (SETQ DOTSEEN NIL)
                  (SETQ END (SUB1 END]
              (COND
                 ((GEQ END START)
                  (SETQ NAME (SUBSTRING FILENAME START END]
          (if (AND SEMI (NEQ SEMI NC))
              then                                           ; Parse version as integer. Note: PARSE-INTEGER demands a
                                                             ; string, but FILENAME might be a symbol.
                   (CL:MULTIPLE-VALUE-SETQ (VERSION END)
                          (CL:PARSE-INTEGER (SUBSTRING FILENAME (ADD1 SEMI))
                                 :JUNK-ALLOWED T))
                   (if (NEQ END (- NC SEMI))
                       then                                  ; Junk found
                            (if (AND PATTERNP (EQ SEMI (SUB1 NC))
                                     (EQ (NTHCHARCODE FILENAME NC)
                                         (CHARCODE *)))
                                then                         ; Version * ok for patterns
                                     (SETQ VERSION '*)
                              else (RETURN NIL))
                     elseif (NOT (AND (> VERSION 0)
                                      (<= VERSION MAX.SMALLP)))
                       then                                  ; Bad version--negative or out of range
                            (RETURN NIL)))
          (RETURN (create NSFILINGPARSE
                          NSDIRECTORIES _ DIRS
                          NSROOTNAME _ NAME
                          NSVERSION _ VERSION
                          NSDIRECTORYP _ (OR (NULL NAME)
                                             (EQ (NCHARS NAME)
                                                 0))
                          NSHASPERIOD _ DOTSEEN])
```

(\NSFILING.ERRORHANDLER

```
  [LAMBDA (STREAM ERRCODE)                                    ; Edited 20-Nov-87 17:03 by bvm:
```

;;; Called when error encountered on STREAM. If STREAM.LOST on an input file, we try to re-establish the connection

```
    (PROG ((PRINTFLG NSFILING.SHOW.STATUS)
           (FAILCNT 0)
           NEWSTREAM HANDLE FULLNAME OLDPTR CON POS)
          (COND
             ((AND (NEQ ERRCODE 'STREAM.LOST)
                    (NEQ ERRCODE 'END))                       ; Not a stream lost type of error.  END can occur if you try to
                                                              ; make a call on a Courier stream at the same time that the other
                                                              ; end decided to time you out.
              (GO EXIT))
             [(NOT (SETQ FULLNAME (fetch FULLFILENAME of STREAM)))
                                                              ; Not a bulk stream with a file in it, maybe in midst of Courier call
              (COND
                 ((SETQ POS (STKPOS (FUNCTION COURIER.EXECUTE.CALL)))
                  (BLOCK 500)
                  ;; Tell courier caller that the stream went away.  Wait a moment for connection process to clean up the mess if there is any
                  (RETFROM POS 'STREAM.LOST T))
                 (T (GO EXIT]
             ((SETQ POS (STKPOS (FUNCTION \COURIER.RESULTS)))    ; Error trying to close the file -- convert this to an error return
              (BLOCK 500)
              (RETFROM POS '(ERROR STREAM.LOST)
                     T))
             ((NEQ (fetch ACCESS of STREAM)
                   'INPUT)                                    ; No help for output files
              (GO EXIT))
             ((NOT (SETQ HANDLE (fetch NSFILING.HANDLE of STREAM)))
                                                              ; Stream already blown away?
              (GO EXIT)))
          (AND PRINTFLG (printout PROMPTWINDOW T "[Reestablishing connection to " FULLNAME " at byte "
                                 (SETQ OLDPTR (GETFILEPTR STREAM))
                                 %,))
      RETRY
          (COND
             ((SETQ NEWSTREAM (\NSFILING.GETFILE (fetch DEVICE of STREAM)
                                        (LET ((ID (fetch NSHFILEID of HANDLE)))
                                             (OR (AND ID (LIST 'FILE.ID ID))
                                                 FULLNAME))
                                        'INPUT
                                        'OLD NIL NIL NIL T))      ; Reopen using ID if possible
              (AND PRINTFLG (printout PROMPTWINDOW "..."))
              (replace SPPERRORHANDLER of (SETQ CON (fetch SPP.CONNECTION of NEWSTREAM))
                 with (FUNCTION ERROR!))
              (COND
                 ((NLSETQ (SETFILEPTR NEWSTREAM OLDPTR))      ; Succeeded in advancing file ptr

                  )
                 ((GREATERP (add FAILCNT 1)
                            3)
                  (GO FAIL))
                 (T (AND PRINTFLG (printout PROMPTWINDOW "failed, retrying "))
                    (GO RETRY)))
              (replace SPPERRORHANDLER of CON with (FUNCTION \NSFILING.ERRORHANDLER))
              (UNINTERRUPTABLY                                ; Smash new stream into old
                  (replace F1 of STREAM with (fetch F1 of NEWSTREAM))
                  (replace F2 of STREAM with (fetch F2 of NEWSTREAM))
                  (replace F3 of STREAM with (fetch F3 of NEWSTREAM))
                  (replace F4 of STREAM with (fetch F4 of NEWSTREAM))
                  (replace F5 of STREAM with (fetch F5 of NEWSTREAM))
                  (replace FW6 of STREAM with (fetch FW6 of NEWSTREAM))
                  (replace FW7 of STREAM with (fetch FW7 of NEWSTREAM))
                  (replace SPPSUBSTREAM of CON with STREAM)
                  (replace CBUFPTR of STREAM with (fetch CBUFPTR of NEWSTREAM))
                  (replace CBUFSIZE of STREAM with (fetch CBUFSIZE of NEWSTREAM))
                  (replace COFFSET of STREAM with (fetch COFFSET of NEWSTREAM)))
              (AND PRINTFLG (printout PROMPTWINDOW "done.]"))
              (RETURN T)))
      FAIL
          (AND PRINTFLG (printout PROMPTWINDOW "...failed.]"))
      EXIT
          (RETURN (\SPP.DEFAULT.ERRORHANDLER STREAM ERRCODE])
```

(\NSFILING.WHENCLOSED

```
  [LAMBDA (STREAM)                                            ; Edited  2-Jun-87 18:42 by bvm:
```

;;; Called when Courier STREAM is closed, by whatever means

```
    (for SESSION in \NSFILING.ACTIVE.SESSIONS bind STREAMPAIRS DEV
       thereis (for PAIR in (SETQ STREAMPAIRS (fetch FSCOURIERSTREAMS of SESSION)) when (EQ (CAR PAIR)
                                                                                          STREAM)
                  do (replace FSCOURIERSTREAMS of SESSION with (DREMOVE PAIR STREAMPAIRS))
```

```
                (COND
                   ((AND (SETQ DEV (\GETDEVICEFROMHOSTNAME (fetch FSDEVICENAME of SESSION)
                                                           T))
                          (fetch (FDEV OPENFILELST) of DEV))
                    (\NSRANDOM.ENSURE.WATCHER DEV)))
                (RETURN T])
```

(\NSFILING.CLOSE.HANDLE

```
[LAMBDA (SESSION HANDLE)                                    ; Edited  5-Jun-87 17:59 by bvm:
```

;; Release the given file handle.

```
(FILING.CALL SESSION 'CLOSE (fetch NSHDATUM of HANDLE)
       SESSION
       'NOERROR])
```

(\NSFILING.FULLNAME

```
[LAMBDA (CONNECTION HANDLE.OR.PARSE VERSION ATOMFLG)        ; Edited 20-Nov-87 18:40 by bvm:
  (PROG (FILENAME DIRLST DIRECTORYFLG FULLNAME PATHNAME FUNNYCHAR DOTSEEN ALREADYQUOTED INFO HANDLE QUOTEDDIRS
          )
        (COND
           ((SETQ INFO (COND
                          ((type? FILINGHANDLE HANDLE.OR.PARSE)
                           (COND
                              ((SETQ FULLNAME (fetch NSHNAME of (SETQ HANDLE HANDLE.OR.PARSE)))
                               (GO EXIT)))
                           (OR (fetch NSHATTRIBUTES of HANDLE)
                               (\NSFILING.FILLIN.ATTRIBUTES CONNECTION HANDLE)))
                          ((LISTP (CADR HANDLE.OR.PARSE))   ; Assume is attribute list itself
                           HANDLE.OR.PARSE)))
            (for PAIR in INFO do (SELECTQ (CAR PAIR)
                                     (IS.DIRECTORY (SETQ DIRECTORYFLG (CADR PAIR)))
                                     (VERSION (SETQ VERSION (CADR PAIR)))
                                     (PATHNAME (SETQ PATHNAME (CADR PAIR)))
                                     NIL))
            [for I from 1 while (<= I NC) bind CH VERS (START _ 1)
                                          (NC _ (NCHARS PATHNAME))
                                          PREVDOT
               do (SELCHARQ (SETQ CH (NTHCHARCODE PATHNAME I))
                        (!                                  ; Version marker
                           (SETQ VERS I))
                        (%'                                 ; quote mark, skip it and next char
                           (add I 1))
                        (/                                  ; Directory marker
                           [push DIRLST (SUBSTRING PATHNAME START (COND
                                                                     ((AND VERS (EQ VERS (- I 2))
                                                                           (EQ (NTHCHARCODE PATHNAME
                                                                                            (ADD1 VERS))
                                                                               (CHARCODE 1)))
                                                                      ; Version 1 in path, toss it out
                                                                       (SUB1 VERS))
                                                                     (T (SUB1 I]
                           (SETQ VERS)
                           (SETQ START (ADD1 I))
                           (SETQ DOTSEEN (SETQ PREVDOT NIL)))
                        (%. (SETQ PREVDOT DOTSEEN)
                            (SETQ DOTSEEN I))
                        ((; %: < > } %])                    ; Funny characters that filing doesn't care about but we do --
                                                            ; need to quote these
                             (SETQ FUNNYCHAR T))
                        NIL)
               finally [SETQ PATHNAME (SUBSTRING PATHNAME START (COND
                                                                   ((NULL VERS)
                                                                    NIL)
                                                                   ((NULL DIRECTORYFLG)
                                                                    ; ordinary file, here's the version
                                                                    (SETQ VERSION (SUBSTRING PATHNAME
                                                                                             (ADD1 VERS)))
                                                                    (SUB1 VERS))
                                                                   ((AND (EQ VERS (- I 2))
                                                                         (EQ (NTHCHARCODE PATHNAME (ADD1 VERS))
                                                                             (CHARCODE 1)))
                                                                    ; Version 1 in path, toss it out
                                                                    (SUB1 VERS]
                  (SETQ FILENAME (COND
                                    (DIRECTORYFLG (SETQ DOTSEEN NIL)
                                           (push DIRLST PATHNAME)
                                           NIL)
                                    ((OR (if (AND DOTSEEN (EQ DOTSEEN (if VERS
                                                                          then (SUB1 VERS)
                                                                        else NC)))
                                             then                ; Ugh--the pathname ended in an actual period, which we usually
                                                                 ; toss out.  I.e. we prefer extensionless files to have no period at
                                                                 ; the end.  So if the server thinks there is one, we'd better say
                                                                 ; FOO'..;1 instead of FOO.;1.
                                         (SETQ DOTSEEN PREVDOT)
```

```
                                                    T)
                                                FUNNYCHAR)          ; May need to quote chars that the server didn't find worth
                                                                    ; quoting.
                                             (\NSFILING.ADDQUOTES PATHNAME T))
                                           (T PATHNAME]

            ;; DIRLST is in reverse order now.

            (for DIR in DIRLST do (push QUOTEDDIRS (COND
                                                       (FUNNYCHAR (\NSFILING.ADDQUOTES DIR T))
                                                       (T DIR))
                                                   '>))
            (SETQ ALREADYQUOTED T)

            ;; Since everything came from a valid (from the server's point of view) pathname, we won't have to add quotes except for characters
            ;; that WE care about (for unpackfilename and friends)

            )
           (T (SETQ FILENAME (fetch NSROOTNAME of HANDLE.OR.PARSE))
              [SETQ QUOTEDDIRS (for DIR in (fetch NSDIRECTORIES of HANDLE.OR.PARSE)
                                  join (LIST (\NSFILING.ADDQUOTES DIR ALREADYQUOTED)
                                             '>]
              (SETQ DIRECTORYFLG (fetch NSDIRECTORYP of HANDLE.OR.PARSE))
              (OR VERSION (SETQ VERSION (fetch NSVERSION of HANDLE.OR.PARSE)))
              (SETQ DOTSEEN (fetch NSHASPERIOD of HANDLE.OR.PARSE))
              (SETQ ALREADYQUOTED T)))
        [SETQ FULLNAME (CONCATLIST (NCONC (LIST '{ (fetch FSNAMESTRING of CONNECTION)
                                                  "}<")
                                          QUOTEDDIRS
                                          (AND (NOT DIRECTORYFLG)
                                               (LIST (OR FILENAME "")
                                                     (COND
                                                        (DOTSEEN ";")
                                                        (T ".;"))
                                                     (OR VERSION ""]
        (COND
           (HANDLE (replace NSHNAME of HANDLE with FULLNAME)))
     EXIT
        (RETURN (COND
                   ((AND ATOMFLG *UPPER-CASE-FILE-NAMES*)      ; Return in 'Lisp file name' form
                    (MKATOM (U-CASE FULLNAME)))
                   (T FULLNAME])

)

;; NSFILING device

(DEFINEQ

(\NSFILING.OPENFILE
  [LAMBDA (FILENAME ACCESS RECOG PARAMETERS DEVICE)              ; Edited 19-Aug-88 17:17 by bvm
    (LET (ATTRIBUTES ATVAL OTHER SEQUENTIAL STREAM)
         (COND
            ((SETQ STREAM (\NSFILING.GETFILE DEVICE (if (NOT (type? STREAM FILENAME))
                                                        then FILENAME
                                                   elseif (OPENED FILENAME)
                                                        then (\ILLEGAL.ARG FILENAME)
                                                   else                ; Reopening a closed stream, such as TEdit might do.
                                                        (fetch FULLFILENAME of FILENAME))
                                     (SELECTQ ACCESS
                                          ((INPUT OUTPUT BOTH APPEND)
                                              ACCESS)
                                          (\ILLEGAL.ARG ACCESS))
                                     (SELECTQ RECOG
                                          ((OLD NEW OLDEST OLD/NEW)  ; explicit recog values
                                              RECOG)
                                          (NIL                       ; Default according to access.  I think maybe the generic system
                                                                     ; does this anyway.
                                               (SELECTQ ACCESS
                                                    (INPUT 'OLD)
                                                    (OUTPUT 'NEW)
                                                    (BOTH 'OLD/NEW)
                                                   NIL))
                                          (\ILLEGAL.ARG RECOG))
                                     NIL
                                     (PROGN                          ; Convert caller's PARAMETERS list to OPENSTREAM to a list
                                                                     ; of filing attributes
                                            [for PAIR in PARAMETERS do (COND
                                                                          [(NLISTP PAIR)
                                                                           (COND
                                                                              ((EQ PAIR 'SEQUENTIAL)
                                                                     ; Obsolete way of asking for sequential access
                                                                               (SETQ SEQUENTIAL T]
                                                                          ((EQ (CAR PAIR)
                                                                               'SEQUENTIAL)
                                                                           (SETQ SEQUENTIAL (CADR PAIR)))
                                                                          ((EQ ACCESS 'INPUT)
                                                                     ; Nothing interesting to do
```

```
                                                                    )
                                                                ([NULL (SETQ ATVAL (
                                                                                \LISP.TO.NSFILING.ATTRIBUTE
                                                                                  (CAR PAIR)
                                                                                  (CADR PAIR]
                                                                                  ; Unrecognized attribute, ignore

                                                                    )
                                                                [(SETQ OTHER (ASSOC (CAR ATVAL)
                                                                                    ATTRIBUTES))
                                                                                  ; Duplicate attribute.  If not consistent, complain
                                                                     (COND
                                                                        ((NOT (EQUAL (CADR OTHER)
                                                                                     (CADR ATVAL)))
                                                                         (ERROR "Inconsistent attributes
                                                                                specified to OPENSTREAM"
                                                                                PARAMETERS]
                                                                (T (push ATTRIBUTES ATVAL]
                                                        [COND
                                                           ((AND (NEQ ACCESS 'INPUT)
                                                                 DEFAULTFILETYPE
                                                                 (NOT (ASSOC 'FILE.TYPE ATTRIBUTES)))
                                                                                  ; If no type specified, use default
                                                            (push ATTRIBUTES `(FILE.TYPE ,(OR (\FILETYPE.FROM.TYPE
                                                                                                DEFAULTFILETYPE)
                                                                                              \NSFILING.TYPE.BINARY]
                                                        ATTRIBUTES)
                                                    NIL SEQUENTIAL))
```

;; Register stream manually in the main device so that there is only one place to look, independent of whether the stream itself uses the
;; random or sequential device

```
(push (fetch (FDEV OPENFILELST) of DEVICE)
      STREAM)
STREAM])
```

(\NSFILING.HANDLE.ERROR

```
[LAMBDA (SESSION ERROR FILENAME)                                    ; Edited  8-Dec-87 12:42 by bvm:
  (if ERROR
      then (PRINTOUT PROMPTWINDOW T (CADR ERROR)
                  "--"
                  (CADDR ERROR)))
  (WITHOUT.SESSION.MONITOR SESSION (CL:ERROR (COND
                                             ((AND (EQ (CADR ERROR)
                                                       'ACCESS.ERROR)
                                                   (STRPOS "ACCESS" (CADDR ERROR)
                                                           NIL NIL NIL NIL UPPERCASEARRAY))
                                              'XCL:FS-PROTECTION-VIOLATION)
                                             (T 'XCL:FILE-WONT-OPEN))
                                        :PATHNAME FILENAME])
```

(\NSFILING.CLOSEFILE

```
[LAMBDA (FILESTREAM OPTIONS)                                        ; Edited 18-Apr-88 13:53 by bvm
  (PROG ((ABORTFLG (LISTGET ':ABORT OPTIONS))
         NEWHANDLE HANDLE SESSION INFO)
        (\GENERIC-UNREGISTER-STREAM (fetch DEVICE of FILESTREAM)
              FILESTREAM)
        (COND
           ((NOT (SETQ SESSION (fetch NSFILING.CONNECTION of FILESTREAM)))
            (GO EXIT)))
```

;; Get the handle from the result of the STORE (for OUTPUT) or from the handle already given to RETRIEVE or REPLACE

```
        (SETQ NEWHANDLE (\BULK.DATA.CLOSE FILESTREAM ABORTFLG))
        (\NSFILING.RELEASE.BULKSTREAM SESSION FILESTREAM)           ; Courier stream now available for use by others
        (COND
           ((SETQ HANDLE (fetch NSFILING.HANDLE of FILESTREAM))
            (\NSRANDOM.RELEASE.HANDLE FILESTREAM)))
        [COND
           [(EQ (CAR NEWHANDLE)
                'ERROR)
            (COND
               ((AND (DIRTYABLE FILESTREAM)
                     (NOT ABORTFLG))
                (ERROR (CONCAT "CLOSEF: File not written
                               " (CADR NEWHANDLE)
                               " -- "
                               (CADDR NEWHANDLE))
                       (fetch FULLFILENAME of FILESTREAM]
           ((OR HANDLE NEWHANDLE)
            [COND
               (NEWHANDLE (SETQ HANDLE (\NSFILING.ADD.TO.CACHE SESSION (create FILINGHANDLE
                                                                               NSHDATUM _ NEWHANDLE
                                                                               NSHNAME _ (fetch FULLFILENAME
                                                                                            of FILESTREAM]
            (COND
```

```
                ((SETQ INFO (fetch NSFILING.NEW.ATTRIBUTES of FILESTREAM))
                 ;; Caller of OPENFILE specified new attributes for this file, so change them now that we've stored the file
                 (if (fetch NSHATTRIBUTES of HANDLE)
                     then                                          ; If attrs are cached, fix them
                          (\NSFILING.UPDATE.ATTRIBUTES HANDLE INFO))
                 (FILING.CALL SESSION 'CHANGE.ATTRIBUTES (fetch NSHDATUM of HANDLE)
                        INFO SESSION 'RETURNERRORS]
      EXIT

     ;; just return
     ])
```

(\NSFILING.EVENTFN

```
  [LAMBDA (DEVICE EVENT)                                           ; Edited 30-Nov-87 13:18 by bvm:
    (SELECTQ EVENT
        (BEFORELOGOUT (for S in (fetch (FDEV OPENFILELST) of DEVICE) when (NEQ (fetch (STREAM DEVICE) of S)
                                                                              DEVICE)
                         do                                        ; Force output on random streams, flush page cache
                            (\CLEARMAP S))                         ; Dispose of any open sessions.
                      (\NSFILING.CLOSE.CONNECTIONS DEVICE :TEST))
        ((AFTERLOGOUT AFTERSAVEVM AFTERMAKESYS AFTERSYSOUT)
             (\NSFILING.CLOSE.CONNECTIONS DEVICE :ABORT)
             [for S in (APPEND (fetch (FDEV OPENFILELST) of DEVICE))
                do (COND
                       ((AND (EQ (fetch (STREAM DEVICE) of S)
                                 DEVICE)
                             (DIRTYABLE S))                        ; Files open for sequential write cannot be recovered.  For now
                                                                   ; we also don't recover input files.
                        (PRINTOUT T T "***Warning: sequential " (COND
                                                                   ((DIRTYABLE S)
                                                                    "output to")
                                                                   (T "input from"))
                               " the file "
                               (fetch FULLFILENAME of S)
                               " has been aborted and cannot be resumed." T T)
                        (CLOSEF S))
                       (T                                          ; Let other streams recover if and when anyone touches them.
                          ]
             (COND
                ((NULL (fetch (FDEV OPENFILELST) of DEVICE))       ; If no open files, dispose of the device
                 [LET [(RANDEVICE (fetch NSRANDOMDEVICE of (fetch DEVICEINFO of DEVICE]
                      (COND
                         (RANDEVICE                                ; Have to break this circularity
                                    (replace DEVICEINFO of RANDEVICE with NIL]
                 (\REMOVEDEVICE DEVICE))))
        NIL])
```

(\NSFILING.DELETEFILE

```
  [LAMBDA (FILENAME DEVICE)                                        ; Edited  8-Dec-87 15:40 by bvm:
    (\NSFILING.GETFILE DEVICE FILENAME 'NONE 'OLDEST 'HANDLE [FUNCTION (LAMBDA (SESSION HANDLE)
                                                                       (COND
                                                                          ((OR (NEQ (fetch NSHBUSYCOUNT
                                                                                       of HANDLE)
                                                                                    0)
                                                                               (\NSFILING.CONFLICTP DEVICE
                                                                                      SESSION HANDLE
                                                                                      'OUTPUT))
                                                                           ; File is in use, can't delete
                                                                           NIL)
                                                                          ((AND (fetch NSHDIRECTORYP
                                                                                   of HANDLE)
                                                                                (NOT (\NSFILING.CHILDLESS-P
                                                                                      SESSION HANDLE)))
                                                                           ; Is a directory with children, can't delete
                                                                           NIL)
                                                                          ((FILING.CALL SESSION 'DELETE
                                                                                  (fetch NSHDATUM of HANDLE)
                                                                                  SESSION
                                                                                  'RETURNERRORS)
                                                                           ; Failed to delete it
                                                                           NIL)
                                                                          (T
                                                                           ; Delete succeeded, handle now invalid
                                                                              (replace FSCACHEDHANDLES
                                                                                 of SESSION
                                                                                 with (DREMOVE HANDLE
                                                                                              (fetch
                                                                                                  FSCACHEDHANDLES
                                                                                                of SESSION)))
                                                                              (\NSFILING.FULLNAME SESSION HANDLE
                                                                                     NIL T]
          T])
```

(\NSFILING.CHILDLESS-P

```
[LAMBDA (SESSION HANDLE)                                    ; Edited  8-Dec-87 15:40 by bvm:
  ;; True if we can tell for sure that directory HANDLE has no children.  Errors return nil
  (EQ [CADR (ASSOC 'NUMBER.OF.CHILDREN (FILING.CALL SESSION 'GET.ATTRIBUTES (fetch NSHDATUM of HANDLE)
                                                [CONSTANT (\FILING.ATTRIBUTE.TYPE.SEQUENCE '(NUMBER.OF.CHILDREN]
                                               SESSION
                                               'NOERROR]
      0])
```

(\NSFILING.DIRECTORYNAMEP

```
[LAMBDA (HOST/DIR DEVICE CREATE?)                           ; Edited  4-May-87 17:21 by bvm:
  ;; Returns T or NIL according to whether or not HOST/DIR is a valid host/directory specification.
  (\NSFILING.GETFILE DEVICE HOST/DIR 'NONE NIL 'DIRECTORY (COND
                                                             (CREATE? :ASK])
```

(\NSFILING.HOSTNAMEP

```
[LAMBDA (HOST DEVICE)                                       ; Edited 11-Jun-87 14:49 by bvm:
  (LET ((SERVER (AND (STRPOS ":" HOST)
                     (LOOKUP.NS.SERVER HOST NIL T)))
        FILINGNAME FULLHOSTNAME)                            ; To avoid useless lookups of PUP names, require
                                                            ; Clearinghouse names to have a colon.
       (COND
          ((NOT SERVER)
           NIL)
          ((\GETDEVICEFROMNAME [SETQ FULLHOSTNAME (MKATOM (U-CASE (NSNAME.TO.STRING (fetch NSFSPARSEDNAME
                                                                                       of SERVER)
                                                                                    T]
                 T T))
          (T (SETQ FILINGNAME (PACK* (fetch NSOBJECT of (fetch NSFSPARSEDNAME of SERVER))
                                     " Filing"))
             [\DEFINEDEVICE FULLHOSTNAME
                    (SETQ DEVICE (create FDEV
                                         using \SPP.BULKDATA.DEVICE DEVICENAME _ FULLHOSTNAME REMOTEP _ T
                                               SUBDIRECTORIES _ T OPENFILE _ (FUNCTION \NSFILING.OPENFILE)
                                               REOPENFILE _ (FUNCTION NILL)
                                               CLOSEFILE _ (FUNCTION \NSFILING.CLOSEFILE)
                                               GETFILEINFO _ (FUNCTION \NSFILING.GETFILEINFO)
                                               SETFILEINFO _ (FUNCTION \NSFILING.SETFILEINFO)
                                               GETEOFPTR _ (FUNCTION \NSFILING.GETEOFPTR)
                                               DELETEFILE _ (FUNCTION \NSFILING.DELETEFILE)
                                               HOSTNAMEP _ (FUNCTION NILL)
                                               GETFILENAME _ (FUNCTION \NSFILING.GETFILENAME)
                                               DIRECTORYNAMEP _ (FUNCTION \NSFILING.DIRECTORYNAMEP)
                                               GENERATEFILES _ (FUNCTION \NSFILING.GENERATEFILES)
                                               RENAMEFILE _ (FUNCTION \NSFILING.RENAMEFILE)
                                               EVENTFN _ (FUNCTION \NSFILING.EVENTFN)
                                               OPENP _ (FUNCTION \GENERIC.OPENP)
                                               REGISTERFILE _ (FUNCTION NILL)
                                               UNREGISTERFILE _ (FUNCTION NILL)
                                               BREAKCONNECTION _ (FUNCTION BREAK.NSFILING.CONNECTION)
                                               DEVICEINFO _ (create NSFILINGDEVICEINFO
                                                                    NSFILESERVER _ SERVER
                                                                    NSFILINGLOCK _ (CREATE.MONITORLOCK FILINGNAME)
                                                                    NSFILINGNAME _ FILINGNAME
                                                                    NSCONNECTIONS _ NIL]
             DEVICE])
```

(\NSFILING.GETFILENAME

```
[LAMBDA (NAME RECOG DEVICE)                                 ; Edited  4-May-87 17:21 by bvm:
  ;; Returns full file name of file or NIL if not found.
  (\NSFILING.GETFILE DEVICE NAME 'NONE RECOG 'NAME])
```

(\NSFILING.GETFILEINFO

```
[LAMBDA (STREAM ATTRIBUTE DEVICE)                           ; Edited  5-May-87 13:12 by bvm:
  (LET (DESIREDPROPS INFO HANDLE)
       (DECLARE (SPECVARS DESIREDPROPS))                    ; Used by \NSFILING.GET.ATTRIBUTES
       (COND
          ((EQ ATTRIBUTE 'ALL)
           (SETQ DESIREDPROPS \NSFILING.ALL.ATTRIBUTE.TYPES)
           (\NSFILING.GET/SETINFO DEVICE STREAM (FUNCTION \NSFILING.GET.ATTRIBUTES)))
          ((NULL (SETQ DESIREDPROPS (\FILING.ATTRIBUTE.TYPE (OR (CADR (ASSOC ATTRIBUTE
                                                                             \LISP.TO.NSFILING.ATTRIBUTES))
                                                                ATTRIBUTE)
                                            T)))
           NIL)
          [(AND [EQ DESIREDPROPS (CONSTANT (\FILING.ATTRIBUTE.TYPE 'SIZE.IN.BYTES]
                (type? STREAM STREAM)
                (LET [(LEN (COND
                             ((fetch RANDOMACCESSP of DEVICE)   ; We know for sure
                              (GETEOFPTR STREAM))
```

```
                                ((DIRTYABLE STREAM)                 ; sequential output stream's length is current fileptr
                                 (GETFILEPTR STREAM]
                            (AND LEN (SELECTQ ATTRIBUTE
                                         (SIZE (FOLDHI LEN BYTESPERPAGE))
                                         LEN]
                    (T [SETQ INFO (COND
                                      ((NOT (MEMB DESIREDPROPS \NSFILING.USEFUL.ATTRIBUTE.TYPES))
                                                                    ; Need to fetch this attribute explicitly
                                       (SETQ DESIREDPROPS (LIST DESIREDPROPS))
                                       (\NSFILING.GET/SETINFO DEVICE STREAM (FUNCTION \NSFILING.GET.ATTRIBUTES)))
                                      ((NOT (type? STREAM STREAM))  ; Not an open stream, so have to look it up
                                       (\NSFILING.GETFILE DEVICE STREAM 'NONE 'OLD 'ATTRIBUTES))
                                      ((NULL (SETQ HANDLE (fetch NSFILING.HANDLE of STREAM)))
                                                                    ; Open for output, don't know attributes yet
                                       NIL)
                                      ((fetch NSHATTRIBUTES of HANDLE))
                                      (T                            ; Stream open but its attributes wiped--retrieve them again
                                         (\NSFILING.FILLIN.ATTRIBUTES (fetch NSFILING.CONNECTION of STREAM)
                                                 HANDLE]
                         (\NSFILING.GETFILEINFO.FROM.PLIST INFO ATTRIBUTE])
```

(\NSFILING.GET.ATTRIBUTES

```
  [LAMBDA (SESSION HANDLE)                                          ; Edited 1-Jun-87 16:08 by bvm:
```

;;; Fetches the DESIREDPROPS of the file whose HANDLE is open on this CONNECTION

```
    (DECLARE (USEDFREE DESIREDPROPS))
    (FILING.CALL SESSION 'GET.ATTRIBUTES (fetch NSHDATUM of HANDLE)
           DESIREDPROPS SESSION 'RETURNERRORS])
```

(\NSFILING.GETFILEINFO.FROM.PLIST

```
  [LAMBDA (PLIST ATTRIBUTE)                                         (* bvm%: "26-Jun-86 15:36")
    (COND
       (PLIST
        (SELECTQ ATTRIBUTE
             (WRITEDATE (\NSFILING.GDATE (CADR (ASSOC 'MODIFIED.ON PLIST))))
             (READDATE (\NSFILING.GDATE (CADR (ASSOC 'READ.ON PLIST))))
             (CREATIONDATE (\NSFILING.GDATE (CADR (ASSOC 'CREATED.ON PLIST))))
             (SIZE (LET [(LENGTH (CADR (ASSOC 'SIZE.IN.BYTES PLIST]
                        (AND LENGTH (FOLDHI LENGTH BYTESPERPAGE))))
             (AUTHOR (LET [(CHNAME (CADR (ASSOC 'CREATED.BY PLIST]
                          (AND CHNAME (NSNAME.TO.STRING CHNAME))))
             (PROTECTION [LET
                          [(PROT (CADR (ASSOC 'ACCESS.LIST PLIST]
                                                                    ; PROT = ((ENTRIES SEQUENCE) (DEFAULTED BOOLEAN))
                                                                    (* (COND ((COURIER.FETCH (FILING . ACCESS.LIST)
                                                                    DEFAULTED of PROT) (push RESULT "(defaulted)"))))
                           (AND
                            PROT
                            (for ENTRY in (COURIER.FETCH (FILING . ACCESS.LIST)
                                                 ENTRIES of PROT)
                               collect (COND
                                          [(SMALLP (SETQ PROT (CADDR ENTRY)))
                                           `(,(CAR ENTRY)
                                             ,(CADR ENTRY)
                                             ,@(COND
                                                  ([EQ PROT (CONSTANT (APPLY 'LOGOR (for PAIR in
                                                                                       \NSFILING.PROTECTION.BITS
                                                                                       collect (CDR PAIR]
                                                                    ; All bits on
                                                   '(ALL))
                                                  (T (for PAIR in \NSFILING.PROTECTION.BITS collect (CAR PAIR)
                                                        when (BITTEST PROT (CDR PAIR]
                                          (T                        ; Must be some other kind of entry, perhaps new filing
                                             ENTRY])
             (TYPE (\TYPE.FROM.FILETYPE (CADR (ASSOC 'FILE.TYPE PLIST))))
             (FILETYPE (CADR (ASSOC 'FILE.TYPE PLIST)))
             (CADR (ASSOC (OR (CADR (ASSOC ATTRIBUTE \LISP.TO.NSFILING.ATTRIBUTES))
                              ATTRIBUTE)
                          PLIST])
```

(\NSFILING.GDATE

```
  [LAMBDA (DATE)                                                    (* lmm "15-Apr-85 16:16")
    (COND
       ((AND DATE (NOT (EQUAL DATE MIN.FIXP)))
        (GDATE DATE])
```

(\NSFILING.SETFILEINFO

```
  [LAMBDA (NAME.OR.STREAM ATTRIBUTE VALUE DEVICE)                   ; Edited 9-Jun-87 15:17 by bvm:
    (PROG ((ATTR/VAL (\LISP.TO.NSFILING.ATTRIBUTE ATTRIBUTE VALUE))
           RESULT)
          (DECLARE (SPECVARS NAME.OR.STREAM ATTR/VAL))
          [COND
```

```
            ((NULL ATTR/VAL)                                        ; Unsupported attribute
             (RETURN NIL))
            ((AND (EQ (CAR ATTR/VAL)
                      'SIZE.IN.BYTES)
                  (type? STREAM NAME.OR.STREAM))                    ; Changing the length on an open stream requires a little more
                                                                    ; than just changing the attribute
             (RETURN (AND (fetch RANDOMACCESSP of DEVICE)
                          (\NSRANDOM.SETEOFPTR NAME.OR.STREAM (CADR ATTR/VAL]
        [SETQ RESULT (\NSFILING.GET/SETINFO DEVICE NAME.OR.STREAM
                             (FUNCTION (LAMBDA (SESSION HANDLE)
                                       (DECLARE (USEDFREE NAME.OR.STREAM ATTR/VAL))
                                       (COND
                                          ((AND (OR (NOT (type? STREAM NAME.OR.STREAM))
                                                    (NEQ HANDLE (fetch NSFILING.HANDLE of NAME.OR.STREAM)))
                                                (\NSFILING.CONFLICTP DEVICE SESSION HANDLE 'OUTPUT))
                                                                    ; We have a stream open on this file, can't change attributes out
                                                                    ; from under it
                                           NIL)
                                          ((FILING.CALL SESSION 'CHANGE.ATTRIBUTES (fetch NSHDATUM of HANDLE)
                                                  (LIST ATTR/VAL)
                                                  SESSION
                                                  'RETURNERRORS))
                                          (T                        ; Change attributes succeeded.  Update cached attributes.
                                             (\NSFILING.UPDATE.ATTRIBUTES HANDLE (LIST ATTR/VAL))
                                             T]
        (RETURN (COND
                   ((LISTP RESULT)
                    (printout PROMPTWINDOW T (COND
                                                ((type? STREAM NAME.OR.STREAM)
                                                 (fetch FULLFILENAME of NAME.OR.STREAM))
                                                (T NAME.OR.STREAM))
                              " -- "
                              (CADDR RESULT))
                    NIL)
                   (T RESULT])
```

(\NSFILING.GET/SETINFO

```
  [LAMBDA (DEVICE STREAM INFOFN)                                    ; Edited 22-May-87 13:09 by bvm:
    (COND
       [(type? STREAM STREAM)
        (PROG (SESSION RESULT)
          RETRY
              (COND
                 ([AND (SETQ SESSION (fetch NSFILING.CONNECTION of STREAM))
                       (OR [NLISTP (SETQ RESULT (CL:FUNCALL INFOFN (fetch NSFILING.CONNECTION of STREAM)
                                                           (fetch NSFILING.HANDLE of STREAM]
                           (NEQ (CAR RESULT)
                                'ERROR)
                           (NEQ (CADR RESULT)
                                'SESSION.ERROR]
                  (RETURN RESULT)))
              (COND
                 ((fetch RANDOMACCESSP of DEVICE)                   ; Get new session
                  (\NSRANDOM.REESTABLISH STREAM)
                  (GO RETRY))
                 (T                                                 ; Sequential stream that was lost.  Hmm.  Just punt out to the file
                                                                    ; name itself
                    (\NSFILING.GETFILE DEVICE (fetch FULLFILENAME of STREAM)
                           'NONE
                           'OLD
                           'HANDLE INFOFN T]
       (T (\NSFILING.GETFILE DEVICE STREAM 'NONE 'OLD 'HANDLE INFOFN T])
```

(\NSFILING.UPDATE.ATTRIBUTES

```
  [LAMBDA (HANDLE NEWATTRS)                                         ; Edited  9-Jun-87 22:11 by bvm:
    ;; Update HANDLE's attribute cache with the set of possibly changed NEWATTRS.  Return the new attribute cache.
    (replace NSHATTRIBUTES of HANDLE with (NCONC [for X in NEWATTRS collect X
                                                    unless (PROGN   ; Don't cache attributes that are in a different form, or that could
                                                                    ; easily change without our knowledge
                                                                   (MEMB (CAR X)
                                                                         '(ACCESS.LIST DEFAULT.ACCESS.LIST
                                                                                 NUMBER.OF.CHILDREN]
                                                 (for X in (fetch NSHATTRIBUTES of HANDLE) collect X
                                                    unless (ASSOC (CAR X)
                                                                  NEWATTRS])
```

(\NSFILING.GETEOFPTR

```
  [LAMBDA (STREAM)                                                  ; Edited 11-Jun-87 14:42 by bvm:
    (COND
       ((DIRTYABLE STREAM)                                          ; Open for output, must be at eof
        (GETFILEPTR STREAM))
       (T                                                           ; Not randaccessp, but we can fake it with the length server gave
                                                                    ; us on opening
```

(\NSFILING.GETFILEINFO STREAM 'LENGTH (fetch DEVICE of STREAM])

(\NSFILING.GENERATEFILES

```
[LAMBDA (DEVICE PATTERN DESIREDPROPS OPTIONS)                    ; Edited 28-Jan-94 19:15 by bvm
    ;; Device method for file enumeration.  Return a generator that enumerates files matching PATTERN.  DESIREDPROPS is set of attributes caller
    ;; may ask for.  If OPTIONS includes RESETLST, caller promises to be wrapped in a RESETLST that we can use to kill an aborted bulk listing.
    (LET
     (SESSION BULKSTREAM RESULT)                                 ; Need these outside of scope of RESETLST in order to process
                                                                 ; the RESETLST option.
     (RESETLST                                                   ; Need RESETLST for \getfilingconnection
         [PROG ((FILING5 T))
           TOP (RETURN (PROG ((PARSE (\NSFILING.PARSE.FILENAME PATTERN T))
                              NAME VERSION DIRPATH DIR N FILTERNEEDED PATHREQUIRED FILTERLIST SCOPELIST
                              INFINITE.DEPTH HANDLE VERSIONFILTER RETURNPROPS)
                             (if [OR (NULL PARSE)
                                     (NULL (OR SESSION (SETQ SESSION (\GETFILINGCONNECTION DEVICE]
                                 then (RETURN NIL))
                             (if (AND FILING5 (NEQ (fetch FSPROTOCOLNAME of SESSION)
                                                   'FILING))
                                 then (SETQ FILING5 NIL))
                             [for TAIL on (SETQ DIRPATH (fetch NSDIRECTORIES of PARSE))
                                when [SETQ N (STRPOS '* (SETQ DIR (CAR TAIL]
                                do                               ; Wildcard in directory part, e.g., <foo>b*r>baz.  By Lisp's rules,
                                                                 ; we want to include <foo>b>r>baz but not <foo>barbaz.tedit.
                                   (if FILING5
                                       then                      ; New filing lets us say ** to match arbitrary components in
                                                                 ; pathname
                                            (SETQ PATHREQUIRED T)
                                            (RPLACA TAIL (\NSFILING.GENERATE.STARS DIR))
                                     else                        ; This is hard.  Get as far down in the tree as possible, then
                                                                 ; enumerate everything
                                          [SETQ FILTERNEEDED (SETQ DIRPATH (for D in DIRPATH collect D
                                                                              until (EQ D DIR]
                                          (SETQ NAME (if (NEQ N 1)
                                                         then    ; If asked to enumerate <foo>b*r>baz, we can at least
                                                                 ; enumerate <foo>b* and filter the rest
                                                              (SUBSTRING DIR 1 N)))
                                          (RETURN))
                                finally ;; Directories are fine, so all the matching happens on the name
                                        (if (STREQUAL (SETQ NAME (fetch NSROOTNAME of PARSE))
                                                      "*.*")
                                            then                 ; Trivial match
                                                 (SETQ NAME NIL)
                                          else (if (STRPOS ".*" NAME -2 NIL T)
                                                   then          ; If name is foo.*, need to enumerate foo* in order to include
                                                                 ; extensionless foo
                                                        (if (NEQ (NTHCHARCODE (SETQ NAME (SUBSTRING NAME 1 -3))
                                                                              -1)
                                                                 (CHARCODE *))
                                                            then (SETQ NAME (CONCAT NAME "*"))
                                                                 ; foo*.* is ok as foo*, but foo.* needs filtering of foo*
                                                                 (SETQ FILTERNEEDED T))
                                                 elseif (EQ (NTHCHARCODE NAME -1)
                                                            (CHARCODE "."))
                                                   then          ; If have explicitly null extension, remove period and filter -- ns
                                                                 ; file server doesn't understand "extension"
                                                        (SETQ NAME (SUBSTRING NAME 1 -2))
                                                        (SETQ FILTERNEEDED T))
                                                 (if (AND FILING5 (SETQ N (STRPOS "*" NAME)))
                                                     then
                                                   ;; Interior * needs to be replaced with ** so that server will match subdirectories along the path.
                                                   ;; May only work in version 5 (Services 10)
                                                        (SETQ NAME (\NSFILING.GENERATE.STARS NAME))
                                                        (SETQ PATHREQUIRED T]
                             (if (NULL (SETQ HANDLE (\NSFILING.CONNECT SESSION (if PATHREQUIRED
                                                                                   then
                                                                                 ; get root directory
                                                                                      NIL
                                                                                   else DIRPATH)
                                                                               T)))
                                 then (RETURN NIL))
                             [SETQ RETURNPROPS
                              (CL:REMOVE-DUPLICATES (APPEND [CONSTANT (\FILING.ATTRIBUTE.TYPE.SEQUENCE
                                                                       '(PATHNAME IS.DIRECTORY]
                                                            (for PROP in DESIREDPROPS
                                                               when (SETQ PROP
                                                                          (\FILING.ATTRIBUTE.TYPE
                                                                           (OR (CADR (ASSOC PROP
                                                                                            \LISP.TO.NSFILING.ATTRIBUTES
                                                                                            ))
                                                                               PROP)
                                                                           T))
                                                               collect PROP]
```

```
                                        ; make sure there are no duplicates, since File server can object
                                        ; to that
[if PATHREQUIRED
    then                                ; Match a full path name
         [push FILTERLIST `(MATCHES (PATHNAME , (\NSFILING.COMPOSE.PATHNAME
                                                 DIRPATH
                                                 (OR NAME '*]
  elseif (NULL NAME)
    then                                ; Enumerate everything
  elseif (STRPOS '* NAME)
    then ;; The following doesn't quite work in Services 8 because the fileserver won't match against
         ;; subdirectory names.
         [push FILTERLIST `(MATCHES (NAME ,NAME]
  else                                  ; Only enumerate versions.
       (push FILTERLIST `(= ,(COURIER.CREATE (FILING . FILTER.ATTRIBUTE)
                                             ATTRIBUTE _ (LIST 'NAME NAME)
                                             INTERPRETATION _ 'STRING]
(SETQ VERSION (fetch NSVERSION of PARSE))
[if (NEQ VERSION '*)
    then                                ; An interesting version -- either a specific one, or none, meaning
                                        ; highest
                                        ; Highest version matching seems not to work in Services 8
         (push FILTERLIST (SETQ VERSIONFILTER
                                `(= ,(COURIER.CREATE (FILING . FILTER.ATTRIBUTE)
                                                     ATTRIBUTE _ (LIST 'VERSION (OR VERSION
                                                                                    \NSFILING.HIGHEST.VERSION
                                                                                    ))
                                                     INTERPRETATION _ 'CARDINAL]
[if (AND FILING.ENUMERATION.DEPTH DIRPATH)
    then
         ;; Controls how many levels in hierarchy to show.  If FILING.ENUMERATION.DEPTH is infinite, then
         ;; let's also ignore the 'files' that are subdirectories
         (push SCOPELIST `(DEPTH ,(OR (SMALLP FILING.ENUMERATION.DEPTH)
                                      (PROGN (SETQ INFINITE.DEPTH T)
                                             65535]
[if FILTERLIST
    then (push SCOPELIST (LIST 'FILTER (if (CDR FILTERLIST)
                                           then (LIST 'AND FILTERLIST)
                                         else (CAR FILTERLIST]
(PROG NIL
  RETRY
       (SETQ BULKSTREAM (FILING.CALL SESSION 'LIST (fetch NSHDATUM of HANDLE)
                                     RETURNPROPS SCOPELIST NIL SESSION 'RETURNERRORS
                                     'KEEPSTREAM))
       (if (EQ (CAR (LISTP BULKSTREAM))
               'ERROR)
           then (if (AND (EQUAL (CDR BULKSTREAM)
                                '(SCOPE.VALUE.ERROR Illegal FILTER))
                         VERSIONFILTER
                         (NULL VERSION))
                    then                ; old versions of Services didn't handle filtering on highest
                                        ; version.  Compromise and return ALL versions
                         (LET ((SCOPE (ASSOC 'FILTER SCOPELIST)))
                              [if (EQ (CADR SCOPE)
                                      VERSIONFILTER)
                                  then (SETQ SCOPELIST (DREMOVE SCOPE SCOPELIST))
                                else ; SCOPE = (FILTER (AND filters))
                                     (CL:SETF (CADADR SCOPE)
                                              (DREMOVE VERSIONFILTER (CADADR SCOPE]
                              (SETQ VERSIONFILTER NIL)
                              (GO RETRY))
                  elseif [AND FILING5 (EQUAL (CDR BULKSTREAM)
                                             '(SCOPE.VALUE.ERROR Unimplemented FILTER]
                    then ;; Grumble.  Unix implementation of filing5 doesn't support * in pathname
                         (SETQ FILING5 NIL)
                         (GO TOP)))
       (if (STREAMP BULKSTREAM)
           then (SETQ RESULT
                      (create FILEGENOBJ
                              NEXTFILEFN _ (FUNCTION \NSFILING.NEXTFILE)
                              FILEINFOFN _ (FUNCTION \NSFILING.FILEINFOFN)
                              GENFILESTATE _
                              (create \NSFILING.GENFILESTATE
                                      NSGENERATOR _ (BULKDATA.GENERATOR BULKSTREAM
                                                                        (fetch FSPROTOCOLNAME of SESSION)
                                                                        'ATTRIBUTE.SEQUENCE)
                                      NSFILTER _ (AND FILTERNEEDED (DIRECTORY.MATCH.SETUP
                                                                    PATTERN))
                                      NSCONNECTION _ SESSION
                                      NSIGNOREDIRECTORIES _ INFINITE.DEPTH
                                      NSBULKSTREAM _ BULKSTREAM)))
         else (if (AND (LISTP BULKSTREAM)
                       (EQ (pop BULKSTREAM)
                           'ERROR))
```

```
                                                    then (PRINTOUT PROMPTWINDOW T "Can't enumerate " PATTERN " because
" (pop BULKSTREAM)
                                                                                    ": ")
                                                                (MAPRINT BULKSTREAM PROMPTWINDOW))
                                                        (SETQ BULKSTREAM NIL])
```

;; We now have either a bulk data listing stream, or we failed. Outside of the RESETLST, let's arrange to kill the listing stream on error

```
(if (AND RESULT (EQMEMB 'RESETLST OPTIONS))
    then (RESETSAVE NIL (LIST (FUNCTION \NSFILING.CLOSE.BULKSTREAM)
                              SESSION BULKSTREAM)))
(OR RESULT (\NULLFILEGENERATOR])
```

(\NSFILING.GENERATE.STARS

```
[LAMBDA (NAME)                                                          ; Edited 15-Sep-87 13:09 by bvm:
  (bind N while (SETQ N (STRPOS "*" NAME N)) do (SETQ NAME (CONCAT (SUBSTRING NAME 1 N)
                                                                    '*
                                                                    (OR (SUBSTRING NAME (+ N 1))
                                                                        "")))
                                                (SETQ N (+ N 3))
                                                                        ; Skip past the * we found, the * we added, and the next char
                                                                        ; (since if it's a *, we don't care).
    finally (RETURN NAME])
```

(\NSFILING.NEXTFILE

```
[LAMBDA (GENFILESTATE NAMEONLY SCRATCHLIST)                             ; Edited 20-Nov-87 18:34 by bvm:
  (PROG ((GENERATOR (fetch NSGENERATOR of GENFILESTATE))
         (SESSION (fetch NSCONNECTION of GENFILESTATE))
         (FILTER (fetch NSFILTER of GENFILESTATE))
         (IGNOREDIRS (fetch NSIGNOREDIRECTORIES of GENFILESTATE))
         INFO NAME)
    LP   (COND
            ((NULL (SETQ INFO (BULKDATA.GENERATE.NEXT GENERATOR)))
                                                                        ; Generator exhausted, so close the bulkdata.
             (LET ((RESETSTATE NIL))                                    ; normal close
                  (\NSFILING.CLOSE.BULKSTREAM SESSION (fetch NSBULKSTREAM of GENFILESTATE)))
             (RETURN NIL))
            ((AND IGNOREDIRS (CADR (ASSOC 'IS.DIRECTORY INFO)))         ; Skip directory files
             (GO LP)))
         (SETQ NAME (\NSFILING.FULLNAME SESSION INFO))
         (COND
            ((AND FILTER (NOT (DIRECTORY.MATCH FILTER NAME)))
             (GO LP)))
         (replace CURRENTINFO of GENFILESTATE with INFO)
         (RETURN (COND
                    (NAMEONLY (NAMEFIELD NAME T))
                    (T NAME])
```

(\NSFILING.FILEINFOFN

```
[LAMBDA (GENFILESTATE ATTRIBUTE)                                        (* bvm%: " 1-May-84 14:04")
  (\NSFILING.GETFILEINFO.FROM.PLIST (fetch CURRENTINFO of GENFILESTATE)
         ATTRIBUTE])
```

(\NSFILING.RENAMEFILE

```
[LAMBDA (DEVICE OLDNAME NEWDEVICE NEWNAME)                              ; Edited  8-Dec-87 20:05 by bvm:
  (COND
     ((EQ (fetch OPENFILE of NEWDEVICE)
          (FUNCTION \NSFILING.OPENFILE))
      (\NSFILING.COPY/RENAME DEVICE OLDNAME NEWDEVICE NEWNAME))
     (T                                                                 ; Different devices, can't rename cleverly.  Ideally we should
                                                                        ; make sure that oldname is deletable, but what follows is at least
                                                                        ; not worse than the old behavior
        (\GENERIC.RENAMEFILE DEVICE OLDNAME NEWDEVICE NEWNAME])
```

(\NSFILING.COPYFILE

```
[LAMBDA (DEVICE FROMFILE NEWDEVICE TOFILE)                              ; Edited  8-Dec-87 17:12 by bvm:
  (COND
     ((EQ (fetch OPENFILE of NEWDEVICE)
          (FUNCTION \NSFILING.OPENFILE))
      (\NSFILING.COPY/RENAME DEVICE FROMFILE NEWDEVICE TOFILE T))
     (T                                                                 ; Different devices, can't rename cleverly.  Ideally we should
                                                                        ; make sure that oldname is deletable, but what follows is at least
                                                                        ; not worse than the old behavior
        (\GENERIC.COPYFILE DEVICE FROMFILE NEWDEVICE TOFILE])
```

(\NSFILING.COPY/RENAME

```
[LAMBDA (DEVICE FROMFILE NEWDEVICE TOFILE COPYFLG)                      ; Edited  9-Dec-87 18:18 by bvm:
```

;; Perform a COPY or RENAME (according to whether COPYFLG is T or NIL) of FROMFILE to TOFILE. DEVICE and NEWDEVICE are NS Filing ;; devices, but not necessarily the same.

;; Between NS servers we can do a copy/rename that preserves maximal information. However, there are some unpleasantnesses: if the ;; destination already exists, we have to delete it before starting; as far as errors go, Lisp wants RENAMEFILE to just return NIL, but COPYFILE ;; must error.

```
(RESETLST
    [PROG ((OLDPARSE (\NSFILING.PARSE.FILENAME FROMFILE))
           (NEWPARSE (\NSFILING.PARSE.FILENAME TOFILE))
           SESSION NEWSESSION NEWDIR OLDDIR NEWPARENT HANDLE NEWHANDLE NEWATTRS VERSION NAME RESULT
           SERIALSTREAM OLDHANDLE SAME-DIR-P DEST-UNIQUE-P)
      ;; The preliminary work is all the same--parse the source and destination, get a handle on the source name and the destination directory,
      ;; check to make sure the source isn't busy and the destination doesn't yet exist.
          [COND
             [(NULL OLDPARSE)                                    ; Bad name
              (RETURN (AND COPYFLG (CL:ERROR 'XCL:INVALID-PATHNAME :PATHNAME FROMFILE]
             [(NULL NEWPARSE)
              (RETURN (AND COPYFLG (CL:ERROR 'XCL:INVALID-PATHNAME :PATHNAME TOFILE]
             [[OR (NULL (SETQ SESSION (\GETFILINGCONNECTION DEVICE)))
                  (NULL (SETQ HANDLE (OR (\NSFILING.LOOKUP.CACHE SESSION FROMFILE)
                                         (\NSFILING.OPEN.HANDLE SESSION (\NSFILING.COMPOSE.PATHNAME
                                                                         (fetch NSDIRECTORIES of OLDPARSE)
                                                                         (fetch NSROOTNAME of OLDPARSE)
                                                                         (OR (fetch NSVERSION of OLDPARSE)
                                                                             '+]
                                                                         ; Can't get to server, or can't get handle on FROMFILE
              (RETURN (AND COPYFLG (CL:ERROR 'XCL:FILE-NOT-FOUND :PATHNAME FROMFILE]
             [[OR (AND (NULL COPYFLG)
                       (NEQ (fetch NSHBUSYCOUNT of HANDLE)
                            0))
                  (\NSFILING.CONFLICTP DEVICE SESSION HANDLE (if COPYFLG
                                                                 then 'INPUT
                                                               else 'OUTPUT]
                                                                 ; File is in use
              (RETURN (AND COPYFLG (CL:ERROR 'XCL:FILE-WONT-OPEN :PATHNAME FROMFILE]
             ([NULL (SETQ NEWSESSION (if (EQ DEVICE NEWDEVICE)
                                         then                    ; Same session will do
                                              SESSION
                                       else (\GETFILINGCONNECTION NEWDEVICE]
                                                                 ; Can't get to destination
              (RETURN (AND COPYFLG (CL:ERROR 'XCL:FILE-NOT-FOUND :PATHNAME TOFILE]
          (SETQ NEWDIR (fetch NSDIRECTORIES of NEWPARSE))
          (SETQ VERSION (fetch NSVERSION of NEWPARSE))
          (if (OR VERSION (fetch NSHDIRECTORYP of HANDLE))
              then                                               ; Destination is uniquely specified, down to the version.
                                                                 ; Directories try hard to be version 1.
                   (SETQ DEST-UNIQUE-P T))
          (if (NULL (SETQ NAME (fetch NSROOTNAME of NEWPARSE)))
              then                                               ; Interpret last directory as the name
                   (SETQ NAME (CAR (LAST NEWDIR)))
                   (SETQ NEWDIR (CL:BUTLAST NEWDIR)))
          (if [AND (NULL COPYFLG)
                   (EQ DEVICE NEWDEVICE)
                   (EQ (LENGTH NEWDIR)
                       (LET [(N (LENGTH (SETQ OLDDIR (fetch NSDIRECTORIES of OLDPARSE]
                             (if (fetch NSHDIRECTORYP of HANDLE)
                                 then                            ; Don't count the last directory--it's the "file"
                                      (- N 1)
                               else N)))
                   (for DIR in NEWDIR always (STRING-EQUAL DIR (pop OLDDIR]
              then                                               ; RENAME uses a simpler call in the case where the source and
                                                                 ; destination directories are identical
                   (SETQ SAME-DIR-P T))
          [SETQ NEWATTRS `((NAME ,(\NSFILING.REMOVEQUOTES NAME))
                           ,@(AND VERSION `((VERSION ,VERSION]
          [COND
             ([AND (OR (NOT SAME-DIR-P)
                       DEST-UNIQUE-P)
                   (NULL (SETQ NEWPARENT (\NSFILING.CONNECT NEWSESSION NEWDIR T T]
                                                                 ; Couldn't get handle on destination directory.  Don't bother if we
                                                                 ; don't need this handle (we don't need it for rename on same dir
                                                                 ; unless there is a uniqueness question)
              (RETURN (AND COPYFLG (CL:ERROR 'XCL:FILE-WONT-OPEN :PATHNAME TOFILE]
          [COND
             ((AND DEST-UNIQUE-P (SETQ OLDHANDLE (\NSFILING.OPEN.HANDLE NEWSESSION NEWATTRS NIL 'NOERROR
                                                                        NEWPARENT)))
                                                                 ; Destination already exists, so we'll get a NotUnique error if we
                                                                 ; COPY/MOVE/SERIALIZE directly.
              (if (if (fetch NSHDIRECTORYP of OLDHANDLE)
                      then                                       ; Old directory ok if it has children or we're copying a
                                                                 ; non-directory
                           (OR (NOT (fetch NSHDIRECTORYP of HANDLE))
                               (NOT (\NSFILING.CHILDLESS-P NEWSESSION OLDHANDLE)))
                    else                                         ; Not file to directory, please
                         (fetch NSHDIRECTORYP of HANDLE))
                  then                                           ; Don't try to overwrite
                       (CL:FORMAT PROMPTWINDOW "~%%Destination ~A already exists." TOFILE)
                       (RETURN (AND COPYFLG (CL:ERROR 'XCL:FILE-WONT-OPEN :PATHNAME TOFILE]
          [if (AND (NULL COPYFLG)
                   (OR OLDHANDLE (NEQ DEVICE NEWDEVICE)))
              then                                               ; RENAME case: we are about to do something we'd rather not
                                                                 ; do (delete destination or copy file) if in the end we're not going
```

```
                                                    ; to have delete access to the source, so check now.
            (if (SETQ RESULT (\NSFILING.CHECK.ACCESS SESSION HANDLE 'WRITE))
                then                                ; No access to delete source
                     (RETURN (AND COPYFLG (\NSFILING.HANDLE.ERROR SESSION RESULT FROMFILE]
      [if OLDHANDLE
          then                                      ; To overwrite old file, have to delete current file first
            [if (SETQ RESULT (FILING.CALL NEWSESSION 'DELETE (fetch NSHDATUM of OLDHANDLE)
                                  NEWSESSION
                                  'RETURNERRORS))
                then                                ; Failed to delete it
                     (RETURN (AND COPYFLG (\NSFILING.HANDLE.ERROR NEWSESSION RESULT TOFILE]
                                                    ; Delete succeeded, handle now invalid
            (replace FSCACHEDHANDLES of NEWSESSION with (DREMOVE OLDHANDLE (fetch FSCACHEDHANDLES
                                                                              of NEWSESSION]
      [if (NOT SAME-DIR-P)
          then                                      ; Be sure not to copy protection along with the file.  Only
                                                    ; exception is a same-dir rename.  You might want the protection
                                                    ; to come along, but it's likely to cause confusion.
            (SETQ NEWATTRS
             (APPEND NEWATTRS
                    `((ACCESS.LIST ,(COURIER.CREATE (FILING . ACCESS.LIST)
                                                    ENTRIES _ NIL DEFAULTED _ T))
                      ,@(AND (fetch NSHDIRECTORYP of HANDLE)
                             `((DEFAULT.ACCESS.LIST ,(COURIER.CREATE (FILING . ACCESS.LIST)
                                                                     ENTRIES _ NIL DEFAULTED _ T]
```

;; Ok, we should be ready to do the copy. If it's the same server, can just use the COPY command, else have to serialize and deserialize

```
      [SETQ RESULT
       (if (EQ DEVICE NEWDEVICE)
           then                                     ; Easy case
            (if COPYFLG
                then (FILING.CALL SESSION 'COPY (fetch NSHDATUM of HANDLE)
                                  (fetch NSHDATUM of NEWPARENT)
                                  NEWATTRS NIL SESSION 'RETURNERRORS)
              elseif SAME-DIR-P
                then                                ; Same directories, just change attributes
                     (FILING.CALL SESSION 'CHANGE.ATTRIBUTES (fetch NSHDATUM of HANDLE)
                            NEWATTRS SESSION 'RETURNERRORS)
                else                                ; Move file to new directory and change its name at the same
                                                    ; time
                   (FILING.CALL SESSION 'MOVE (fetch NSHDATUM of HANDLE)
                          (fetch NSHDATUM of NEWPARENT)
                          NEWATTRS SESSION 'RETURNERRORS))
         elseif (SETQ RESULT (\NSFILING.CHECK.ACCESS NEWSESSION NEWPARENT 'ADD))
           then                                     ; No access to write destination
                (RETURN (AND COPYFLG (\NSFILING.HANDLE.ERROR SESSION RESULT FROMFILE)))
           else                                     ; Copy with serialize-deserialize
                (if (TYPENAMEP (SETQ SERIALSTREAM (FILING.CALL SESSION 'SERIALIZE (fetch NSHDATUM
                                                                                     of HANDLE)
                                                          NIL SESSION 'RETURNERRORS 'KEEPSTREAM))
                               'STREAM)
                    then (CL:UNWIND-PROTECT
                             [PROGN (add (fetch NSHBUSYCOUNT of HANDLE)
                                         1)
                                    (RELEASE.MONITORLOCK (fetch FSSESSIONLOCK of SESSION))
                                                    ; we don't need this lock while transferring--don't keep the
                                                    ; session busy
                                    (PROG1 (\NSFILING.DESERIALIZE1 NEWSESSION NEWPARENT NEWATTRS SERIALSTREAM
                                                 (FUNCTION \BULK.DATA.CLOSE))
                                           (if (NOT COPYFLG)
                                               then ; we need to get the source lock back in order to delete the
                                                    ; source.
                                                    (OBTAIN.MONITORLOCK (fetch FSSESSIONLOCK of SESSION))))]
                             ;; Cleanup after the SERIALIZE finishes
                             (add (fetch NSHBUSYCOUNT of HANDLE)
                                  -1)
                             (\BULK.DATA.CLOSE SERIALSTREAM)
                             (\NSFILING.RELEASE.BULKSTREAM SESSION SERIALSTREAM))
                 else (RETURN (AND COPYFLG (\NSFILING.HANDLE.ERROR SESSION HANDLE FROMFILE]
      (RETURN (COND
                 ((NEQ (CAR (LISTP RESULT))
                       'ERROR)                      ; Success--note new file in cache
                  (SETQ NEWHANDLE (if (OR COPYFLG (NEQ DEVICE NEWDEVICE))
                                      then (\NSFILING.ADD.TO.CACHE NEWSESSION (create FILINGHANDLE
                                                                                      NSHDATUM _ RESULT))
                                                    ; In place move invalidates our knowledge about handle
                                      else (replace NSHATTRIBUTES of HANDLE
                                              with (replace NSHNAME of HANDLE with NIL))
                                           HANDLE))
                  [if (AND (NULL COPYFLG)
                           (NEQ DEVICE NEWDEVICE))
                      then                          ; Now have to delete the source
                           (if (SETQ RESULT (FILING.CALL SESSION 'DELETE (fetch NSHDATUM of HANDLE)
                                                 SESSION
                                                 'RETURNERRORS))
                               then                 ; Failed to delete it.  Unclear what we should do about the
```

```
                                                    ; destination at this point.  I planned on not getting this error, so
                                                    ; tell user.  Typical case: I tried to move a directory one of whose
                                                    ; children I do not have delete access to
                                        (RELEASE.MONITORLOCK (fetch FSSESSIONLOCK of SESSION))
                                        (RELEASE.MONITORLOCK (fetch FSSESSIONLOCK of NEWSESSION))
                                                    ; Release locks so not tied up while in error
                                        (RETURN (CL:ERROR "Successfully copied ~A to ~A, but failed to delete
                                                    the source because ~A: ~A." (\NSFILING.FULLNAME
                                                                                    SESSION HANDLE)
                                                    (\NSFILING.FULLNAME NEWSESSION NEWHANDLE)
                                                    (CADR RESULT)
                                                    (CADDR RESULT]
                                  (\NSFILING.FULLNAME NEWSESSION NEWHANDLE))
                                 (COPYFLG                   ; Failure--signal some error
                                        (\NSFILING.HANDLE.ERROR NEWSESSION RESULT TOFILE])])

)

;; Random access methods

(DEFINEQ

(\NSRANDOM.CLOSEFILE
  [LAMBDA (STREAM)                                          ; Edited 20-Nov-87 17:28 by bvm:
    ;; Close method for a stream open on the random access Filing device
    (RESETLST
        (PROG ((SESSION (fetch NSFILING.CONNECTION of STREAM)))
              (if SESSION
                  then ;; We ought not have to do this, but sometimes ill-disciplined folk try to close the same stream twice, by lazily calling
                       ;; CLOSEF? and getting in here while we're talking to the server.  We don't have monitor locks per stream (though we
                       ;; probably should), so use the session's lock.  This is obviously inadequate in general, since the session might have died,
                       ;; but it should handle the average case.
                       (OBTAIN.MONITORLOCK (fetch FSSESSIONLOCK of SESSION)
                              NIL T))
              (if (NULL (fetch (STREAM ACCESS) of STREAM))
                  then                                      ; Somebody else already closed us
                       (RETURN))
              (\CLEARMAP STREAM)                            ; Force out dirty buffers
              (COND
                 ((DIRTYABLE STREAM)                        ; Truncate to current length
                  (\NSRANDOM.TRUNCATEFILE STREAM)))
              (\NSRANDOM.RELEASE.HANDLE STREAM)             ; Release controls
              (\GENERIC-UNREGISTER-STREAM (fetch DEVICEINFO of (fetch DEVICE of STREAM))
                     STREAM)
              (replace (STREAM ACCESS) of STREAM with NIL)))
    STREAM])

(\NSRANDOM.RELEASE.HANDLE
  [LAMBDA (STREAM)                                          ; Edited 20-Nov-87 17:00 by bvm:
    ;; Release STREAM's hold on its file handle.  We also remove the HANDLE and CONNECTION from the stream, etc.
    (LET ((HANDLE (fetch NSFILING.HANDLE of STREAM))
          (SESSION (fetch NSFILING.CONNECTION of STREAM)))
         (replace NSFILING.HANDLE of STREAM with NIL)
         (replace NSFILING.CONNECTION of STREAM with NIL)
         (COND
            ((NULL HANDLE))
            ((NEQ (fetch NSHBUSYCOUNT of HANDLE)
                  1)
             ;; More than one user, so keep controls
             (add (fetch NSHBUSYCOUNT of HANDLE)
                  -1))
            (T (replace NSHBUSYCOUNT of HANDLE with 0)
               (COND
                  ((AND SESSION (fetch NSHACCESS of HANDLE))    ; Release lock held on the handle.  Session may have been
                                                                ; dropped, in which case no need to change control
                   (\NSRANDOM.RELEASE.LOCK SESSION HANDLE])

(\NSRANDOM.RELEASE.LOCK
  [LAMBDA (SESSION HANDLE)                                  ; Edited  3-Jun-87 18:22 by bvm:
    (FILING.CALL SESSION 'CHANGE.CONTROLS (fetch NSHDATUM of HANDLE)
           '((LOCK NONE))
           SESSION
           'RETURNERRORS)
    (replace NSHACCESS of HANDLE with NIL])

(\NSRANDOM.RELEASE.IF.ERROR
  [LAMBDA (SESSION HANDLE)                                  ; Edited 26-Aug-87 15:30 by bvm:
    (AND RESETSTATE (\NSRANDOM.RELEASE.LOCK SESSION HANDLE])
```

(\NSRANDOM.CREATE.STREAM

```
[LAMBDA (SESSION HANDLE DEVICE ACCESS GOTCONTROLS OLDSTREAM CHECKACCESS)
                                                        ; Edited 23-May-2024 23:07 by frank
                                                        ; Edited 19-Aug-88 17:24 by bvm
    (PROG NIL
          [COND
             ((NOT GOTCONTROLS)
              ;; Acquire lock on file for duration of open stream.  Need this so that nobody can get in between calls to RetrieveBytes or
              ;; ReplaceBytes
              (LET ((OLDACCESS (fetch NSHACCESS of HANDLE))
                    ERROR)
                   [COND
                      ((SELECTQ OLDACCESS
                            ((NIL)                      ; Just a cached handle, no controls
                                NIL)
                            (OUTPUT                     ; Handle already open for write, can't do anything else
                                    T)
                            (INPUT                      ; Open for input, so only other input streams allowed.
                                   (NEQ ACCESS 'INPUT))
                            (SHOULDNT))
                       (RETURN (LISPERROR "FILE WON'T OPEN" (\NSFILING.FULLNAME SESSION HANDLE]
                   (COND
                      ((NEQ OLDACCESS 'INPUT)           ; Get a share/exclusive control.  If OLDACCESS is INPUT, we
                                                        ; have already obtained this control
                       (COND
                          ((SETQ ERROR (FILING.CALL SESSION 'CHANGE.CONTROLS (fetch NSHDATUM of HANDLE)
                                                    `[(LOCK ,(SELECTQ ACCESS
                                                                      (INPUT 'SHARE)
                                                                      'EXCLUSIVE]
                                                    SESSION
                                                    'RETURNERRORS))
                           (RETURN ERROR)))
                       (RESETSAVE NIL (LIST (FUNCTION \NSRANDOM.RELEASE.IF.ERROR)
                                            SESSION HANDLE))     ; If this open doesn't succeed, be sure to release this lock.
                       (replace NSHACCESS of HANDLE with (SELECTQ ACCESS
                                                                  ((BOTH APPEND)
                                                                      'OUTPUT)
                                                                  ACCESS]
          [COND
             (CHECKACCESS
                      ;; Problem: How can we tell NOW whether we have access rights to write this file?  At least in the case of a new file, the
                      ;; CREATE procedure will tell us if we had ADD access, but even then we might perversely not have WRITE access.
                      (LET [(ERROR (\NSFILING.CHECK.ACCESS SESSION HANDLE 'WRITE]
                           (AND ERROR (RETURN ERROR]
          (LET* ((ATTRS (OR (fetch NSHATTRIBUTES of HANDLE)
                            (\NSFILING.FILLIN.ATTRIBUTES SESSION HANDLE)))
                 (LEN (CADR (ASSOC 'SIZE.IN.BYTES ATTRS)))
                 S EOF)
                [COND
                   (OLDSTREAM [LET [(OLDATTRS (fetch NSHATTRIBUTES of (fetch NSFILING.HANDLE of OLDSTREAM]
                                   (COND
                                      ([OR (NOT (EQUAL LEN (fetch NSFILING.SERVER.LENGTH of OLDSTREAM)))
                                           (NOT (EQUAL (CADR (ASSOC 'CREATED.ON ATTRS))
                                                       (CADR (ASSOC 'CREATED.ON OLDATTRS]
                                                        ; file has changed!
                                       (\NSRANDOM.STREAM.CHANGED OLDSTREAM HANDLE]
                                                        ; If got here, user let us continue
                               (replace NSFILING.HANDLE of (SETQ S OLDSTREAM) with HANDLE))
                   (T (SETQ EOF (SELECTQ ACCESS
                                         (OUTPUT 0)
                                        LEN))
                      (SETQ S (create STREAM
                                      DEVICE _ DEVICE
                                      EPAGE _ (FOLDLO EOF BYTESPERPAGE)
                                      EOFFSET _ (IMOD EOF BYTESPERPAGE)
                                      MULTIBUFFERHINT _ T))
                      (if (EQ ACCESS 'APPEND)
                          then                          ; File pos at end
                               (freplace (STREAM CPAGE) of S with (fetch (STREAM EPAGE) of S))
                               (freplace (STREAM COFFSET) of S with (fetch (STREAM EOFFSET) of S))
                        else                            ; File pos at start
                               (freplace (STREAM CPAGE) of S with 0)
                               (freplace (STREAM COFFSET) of S with 0]
                   (replace NSFILING.SERVER.LENGTH of S with LEN)
                   (RETURN S])
```

(\NSRANDOM.READPAGES

```
[LAMBDA (STREAM FIRSTPAGE# BUFFERS)                     ; Edited  3-Sep-87 12:03 by bvm:
```

;; Read pages method for NSFIling Random access device.

```
(COND
   ((LISTP BUFFERS)
    (\NSRANDOM.READ.SEGMENT STREAM FIRSTPAGE# BUFFERS))
   (T ;; Single buffer. We special case this because we want to in general fetch several pages at once to improve performance
```

```
        (COND
           ((NULL (fetch NSFILING.CONNECTION of STREAM))          ; Session lost, e.g., after logout.  Want to reestablish stream
                                                                  ; immediately, even if all we're going to do is clear the buffer.
            (\NSRANDOM.REESTABLISH STREAM)))
        [LET ((EP (fetch (STREAM EPAGE) of STREAM))
              (EO (fetch (STREAM EOFFSET) of STREAM))
              CACHE NMORE EXTRABUFFERS)
             (COND
                ((OR (> FIRSTPAGE# EP)
                     (AND (EQ FIRSTPAGE# EP)
                          (EQ EO 0)))                             ; Past eof.  This is silly
                 (\CLEARBYTES BUFFERS 0 BYTESPERPAGE))
                [(SETQ CACHE (\NSRANDOM.FETCH.CACHE STREAM FIRSTPAGE#))
                                                                  ; We fetched it earlier, so this is easy
                 (\BLT BUFFERS (CADR CACHE)
                       WORDSPERPAGE)
                 (COND
                    (\NSRANDOM.CHECK.CACHE (\NSRANDOM.CHECK.CACHE (fetch NSFILING.PAGE.CACHE of STREAM]
                (T                                                ; Have to fetch it.  Get next few pages while we're at it.
                   [COND
                      ((AND (>= FIRSTPAGE# (fetch NSFILING.LAST.REQUEST of STREAM))
                            (PROGN [for I from 1
                                      to (SETQ NMORE
                                           (IMIN *NSFILING-PAGE-CACHE-INCREMENT*
                                                 (- (if (DIRTYABLE STREAM)
                                                        then      ; For output files, it is possible for our local eof to be greater than
                                                                  ; the server's, in which case we'd better not try to read.
                                                            (FOLDLO (SUB1 (fetch NSFILING.SERVER.LENGTH
                                                                           of STREAM))
                                                                    BYTESPERPAGE)
                                                      elseif (EQ EO 0)
                                                        then (SUB1 EP)
                                                      else EP)
                                                    FIRSTPAGE#)))
                                      when (\NSRANDOM.FETCH.CACHE STREAM (+ FIRSTPAGE# I)
                                                                  T)
                                      do
                                   ;; This page already cached, so don't bother fetching it again.  Notice that this algorithm is pessimal
                                   ;; for reading a file backward, but it's hard for me to do better without more knowledge of what's
                                   ;; already buffered in the stream.
                                         (RETURN (SETQ NMORE (SUB1 I]
                                  (NEQ NMORE 0)))
                       ;; Ok, have a range to read.  First check says don't read multiple if going backward in file (I don't know how to do this
                       ;; well--there are many common cases, such as Lafite get mail and backward searches, that would be handled
                       ;; pessimally if I retrieve multiple pages here).
                        (SETQ EXTRABUFFERS (\NSRANDOM.PREPARE.CACHE STREAM NMORE]
                      (\NSRANDOM.READ.SEGMENT STREAM FIRSTPAGE# BUFFERS EXTRABUFFERS NMORE)
                      (COND
                         (\NSRANDOM.CHECK.CACHE (\NSRANDOM.CHECK.CACHE (fetch NSFILING.PAGE.CACHE of STREAM)
                                                                       T]
        (replace NSFILING.LAST.REQUEST of STREAM with FIRSTPAGE#])
```

(\NSRANDOM.READ.SEGMENT

```
  [LAMBDA (STREAM FIRSTPAGE# BUFFERS EXTRABUFFERS NEXTRA)           ; Edited 27-Aug-87 11:30 by bvm:
    ;; Read contents of STREAM starting at FIRSTPAGE# into successive members of BUFFERS.  In the case that BUFFERS is a single buffer, read
    ;; additional NEXTRA pages into page cache entries EXTRABUFFERS.
    (PROG (SESSION)
      RETRY
          (COND
             ((NULL (SETQ SESSION (fetch NSFILING.CONNECTION of STREAM)))
                                                                  ; Session lost, e.g., after logout
              (\NSRANDOM.REESTABLISH STREAM)
              (GO RETRY)))
          (LET* ((EP (fetch (STREAM EPAGE) of STREAM))
                 (EO (fetch (STREAM EOFFSET) of STREAM))
                 [BYTESTOFETCH (COND
                                  [EXTRABUFFERS                   ; Caller assures us that at worst, the last extra buffer is the end
                                                                  ; of file.
                                        (+ (UNFOLD NEXTRA BYTESPERPAGE)
                                           (COND
                                              ((EQ (+ FIRSTPAGE# NEXTRA)
                                                   EP)
                                               EO)
                                              (T BYTESPERPAGE]
                                  (T                              ; Just a single list of buffers
                                     (for BUF inside BUFFERS as PAGE from FIRSTPAGE#
                                        sum (COND
                                               ((< PAGE EP)
                                                BYTESPERPAGE)
                                               ((EQ PAGE EP)
                                                EO)
                                               (T 0]
```

```
                (HANDLE (fetch NSFILING.HANDLE of STREAM))
                BYTES-TIL-EOF)
               (COND
                  [[AND (NEQ BYTESTOFETCH 0)
                        (OR (NOT (DIRTYABLE STREAM))
                            (COND
                               ([> BYTESTOFETCH (SETQ BYTES-TIL-EOF (- (fetch NSFILING.SERVER.LENGTH
                                                                          of STREAM)
                                                                       (UNFOLD FIRSTPAGE# BYTESPERPAGE]
                                                                ; For output files, it is possible for our local eof to be greater than
                                                                ; the server's, in which case we'd better not try to read.
                                (> (SETQ BYTESTOFETCH BYTES-TIL-EOF)
                                   0))
                               (T T]                            ; There is something to retrieve
                   (LET [(ERROR (FILING.CALL SESSION 'RETRIEVE.BYTES (fetch NSHDATUM of HANDLE)
                                             (COURIER.CREATE (FILING . BYTE.RANGE)
                                                    FIRSTBYTE _ (UNFOLD FIRSTPAGE# BYTESPERPAGE)
                                                    COUNT _ BYTESTOFETCH)
                                             [FUNCTION (LAMBDA (BULKSTREAM)
                                                         ;; What to do with the bulk data
                                                         (LET ((PAGENO FIRSTPAGE#)
                                                               (TOTALBYTES BYTESTOFETCH))
                                                                ; Note that we must keep local copy of the number of bytes
                                                                ; expected, since FILING.CALL can iterate (when stream lost).
                                                              (for BUF inside BUFFERS
                                                                 do (COND
                                                                       ((>= TOTALBYTES BYTESPERPAGE)
                                                                       ; Fetch a whole page
                                                                       (\BINS BULKSTREAM BUF 0 BYTESPERPAGE)
                                                                       (SETQ TOTALBYTES (- TOTALBYTES BYTESPERPAGE))
                                                                       )
                                                                       ((> TOTALBYTES 0)
                                                                       ; Fetch remaining bytes of last page
                                                                       (\BINS BULKSTREAM BUF 0 TOTALBYTES)
                                                                       (\CLEARBYTES BUF TOTALBYTES (- BYTESPERPAGE
                                                                                                      TOTALBYTES))
                                                                       (SETQ TOTALBYTES 0))
                                                                       (T
                                                                       ; At end of actual file, so just clear buffer
                                                                          (\CLEARBYTES BUF 0 BYTESPERPAGE)))
                                                                       (add PAGENO 1))
                                                              (from 1 to NEXTRA as PAIR in EXTRABUFFERS
                                                                 do (RPLACA PAIR -1)
                                                                       ; Temporarily make invalid
                                                                    (COND
                                                                       ((>= TOTALBYTES BYTESPERPAGE)
                                                                       ; Fetch a whole page
                                                                       (\BINS BULKSTREAM (CADR PAIR)
                                                                              0 BYTESPERPAGE)
                                                                       (SETQ TOTALBYTES (- TOTALBYTES BYTESPERPAGE))
                                                                       )
                                                                       ((> TOTALBYTES 0)
                                                                       ; Fetch remaining bytes of last page
                                                                       (\BINS BULKSTREAM (CADR PAIR)
                                                                              0 TOTALBYTES)
                                                                       (\CLEARBYTES (CADR PAIR)
                                                                              TOTALBYTES
                                                                              (- BYTESPERPAGE TOTALBYTES))
                                                                       (SETQ TOTALBYTES 0))
                                                                       (T
                                                                       ; This better never happen
                                                                          (HELP "Inconsistency in READPAGE byte
                                                                                 count")))
                                                                    (RPLACA PAIR PAGENO)
                                                                    (add PAGENO 1))
                                                              (COND
                                                                 ((NOT (EOFP BULKSTREAM))
                                                                       ; RetrieveBytes returned more data than we requested.
                                                                  (COURIER.ABORT.BULKDATA '(ERROR TRANSFER.ERROR
                                                                                                  FormatIncorrect]
                                             SESSION
                                             'RETURNERRORS]
                         (COND
                            (ERROR (\NSRANDOM.HANDLE.ERROR ERROR STREAM SESSION 'RETRIEVE.BYTES)
                                   (GO RETRY)))
                         (COND
                            ((NOT (fetch NSHWASREAD of HANDLE))
                             ;; Reading file has changed its  read date.  We assume this happens only once per handle, that the file service
                             ;; does not change the date on every read!
                             (LET [(ATTR (ASSOC 'READ.ON (fetch NSHATTRIBUTES of HANDLE]
                                  [COND
                                     (ATTR (replace NSHATTRIBUTES of HANDLE with (DREMOVE ATTR
                                                                                          (fetch NSHATTRIBUTES
                                                                                             of HANDLE]
                                  (replace NSHWASREAD of HANDLE with T]
```

```
                (T                                              ; Nothing to retrieve, just clear buffers (pmap code ought to catch
                                                                ; this)
                   (for BUF inside BUFFERS do (\CLEARBYTES BUF 0 BYTESPERPAGE])
```

(\NSRANDOM.PREPARE.CACHE

```
[LAMBDA (STREAM NPAGES)                                         ; Edited 10-Jun-87 20:33 by bvm:
  (LET ((CACHE (fetch NSFILING.PAGE.CACHE of STREAM)))
       (COND
          ((NULL CACHE)                                         ; No cache yet, so create one with n pages in it
           [SETQ CACHE (for I from 1 to NPAGES collect (LIST -1 (NCREATE 'VMEMPAGEP]
           (replace NSFILING.PAGE.CACHE of STREAM with (create NSPAGECACHE
                                                               NSPSIZE _ NPAGES
                                                               NSPTAIL _ (LAST CACHE)
                                                               NSPBUFFERS _ CACHE))
           CACHE)
          (T (COND
                (\NSRANDOM.CHECK.CACHE (\NSRANDOM.CHECK.CACHE CACHE)))
             (PROG ((OLDSIZE (fetch NSPSIZE of CACHE))
                    (HEAD (fetch NSPHEADER of CACHE))
                    PREV FREETAIL NAVAIL NCREATED NNEEDED)
              RETRY
                   (SETQ FREETAIL HEAD)                         ; Find first free cache page.  (CDR HEAD) is the first buffer.
                   (while (SETQ FREETAIL (CDR (SETQ PREV FREETAIL))) when (EQ (CAAR FREETAIL)
                                                                             -1)
                      do                                        ; This buffer is free
                         (SETQ NAVAIL 1)
                         [bind PREVFREE (MORETAIL _ FREETAIL) while (SETQ MORETAIL (CDR (SETQ PREVFREE MORETAIL
                                                                                           )))
                            do (COND
                                  ((EQ (CAAR MORETAIL)
                                       -1)
                                   (add NAVAIL 1))
                                  (T                            ; Not all empty's are at end.  Move these there and try again.
                                     (UNINTERRUPTABLY
                                         ;; Want to transform PREV.FREETAIL...PREVFREE.MORETAIL...LAST to be
                                         ;; PREV.MORETAIL...LAST.FREETAIL...PREVFREE
                                         (RPLACD PREV MORETAIL)
                                                                ; Splice out
                                         (RPLACD PREVFREE NIL)
                                         (RPLACD (fetch (NSPAGECACHE NSPTAIL) of CACHE)
                                                FREETAIL)       ; Attach to end of list
                                         (replace (NSPAGECACHE NSPTAIL) of CACHE with PREVFREE)
                                                                ; Update end pointer

                                         )
                                     (GO RETRY]
                         (RETURN)
                      finally                                   ; No free buffers found
                             (SETQ NAVAIL 0))
              ;; There are now NAVAIL free buffers, the first of which is in NEWTAIL
                   [COND
                      [(<= NPAGES NAVAIL)                       ; That's enough, don't need to allocate any
                       (SETQ NCREATED 0)
                       (RPTQ (- NAVAIL NPAGES)                  ; Want to use the LAST n pages in the case where there are
                                                                ; more free pages than we need
                             (SETQ FREETAIL (CDR (SETQ PREV FREETAIL]
                      [(<= (SETQ NNEEDED (- NPAGES NAVAIL))
                           (SETQ NCREATED (- *NSFILING-PAGE-CACHE-LIMIT* OLDSIZE)))
                                                                ; NCREATED (Maximum buffers we can add) is more than we
                                                                ; need, so use free buffers found above plus just what we need
                       (SETQ NCREATED NNEEDED)
                       (COND
                          ((NULL FREETAIL)                      ; All the created buffers get used, no old ones, so they all go on
                                                                ; front.
                           (SETQ FREETAIL (CDR (SETQ PREV HEAD]
                      ((< NPAGES *NSFILING-PAGE-CACHE-LIMIT*)   ; Create as buffers to get up to limit, and additionally use as
                                                                ; many old buffers as needed to get n.
                       (SETQ PREV (CL:NTHCDR (- OLDSIZE (- NPAGES NCREATED))
                                             HEAD))             ; Fast version of (NLEFT Buffers NPAGES-NCREATED)
                       (SETQ FREETAIL (CDR PREV)))
                      (T                                        ; Perverse case--usually increment < limit.  But do it anyway: use
                                                                ; all existing buffers, and allocate enough new ones to satisfy
                                                                ; request.
                         (SETQ NCREATED (- NPAGES OLDSIZE))
                         (SETQ PREV HEAD)
                         (SETQ FREETAIL (CDR PREV]
              ;; Have HEAD-->FIRST...PREV.FREETAIL...LAST and want to turn it into HEAD-->NewBufs.FREETAIL...LAST.FIRST...PREV
                   [to NCREATED do (push FREETAIL (LIST -1 (NCREATE 'VMEMPAGEP]
                                                                ; Create new buffers as needed
                   (UNINTERRUPTABLY                             ; Need to maintain consistency here...
                       (COND
                          ([AND (NEQ PREV HEAD)
                                (NOT (NULL (CDR PREV]           ; There is non-trivial rearrangement to be done.
```

```
                            (RPLACD PREV NIL)                    ; PREV is new end of list
                            (RPLACD (fetch NSPTAIL of CACHE)
                                   (CDR HEAD))                   ; Splice old head onto old last
                            (replace NSPTAIL of CACHE with PREV)  ; PREV is new last

                            ))
                          (RPLACD HEAD FREETAIL)                 ; New buffer list
                          [COND
                             ((NEQ NCREATED 0)
                              (replace NSPSIZE of CACHE with (+ OLDSIZE NCREATED])
                        (RETURN FREETAIL])
```

(\NSRANDOM.FETCH.CACHE

```
  [LAMBDA (STREAM PAGENO KEEP)                                   ; Edited 3-Sep-87 12:03 by bvm:
    (LET ((CACHE (fetch NSFILING.PAGE.CACHE of STREAM)))
         (COND
            (CACHE (LET ((TAIL (fetch (NSPAGECACHE NSPHEADER) of CACHE))
                          PREV PAIR)
                     ;; Cache is constructed so that there is always a header node we can rplacd to change first element of real list.
                     ;; Contents of header node happens to be the pointer to the tail of the list.
                     (while (SETQ TAIL (CDR (SETQ PREV TAIL))) when (EQ (CAR (SETQ PAIR (CAR TAIL)))
                                                                        PAGENO)
                        do                                         ; Found it.
                            (COND
                               ((NOT KEEP)                         ; Removing it from cache, so move node to end of list.
                                [COND
                                   ((CDR TAIL)                     ; Not already at end
                                    (UNINTERRUPTABLY
                                        (RPLACD PREV (CDR TAIL))
                                                                   ; Splice out
                                        (RPLACD TAIL NIL)
                                        (RPLACD (fetch (NSPAGECACHE NSPTAIL) of CACHE)
                                               TAIL)               ; Attach to end of list
                                        (replace (NSPAGECACHE NSPTAIL) of CACHE with TAIL)
                                                                   ; Update end pointer

                                        )]                         ; Mark pair with impossible page number -1
                                    (RPLACA PAIR -1)))
                            (RETURN PAIR])
```

(\NSRANDOM.CHECK.CACHE

```
  [LAMBDA (CACHE CHECKORDER)                                     ; Edited 10-Jun-87 19:21 by bvm:
    (COND
       ((NULL CACHE)                                             ; Empty cache
        NIL)
       ((NEQ (fetch NSPSIZE of CACHE)
             (LENGTH (fetch NSPBUFFERS of CACHE)))
        (HELP "Cache length is wrong"))
       ((NEQ (fetch (NSPAGECACHE NSPTAIL) of CACHE)
             (LAST (fetch NSPBUFFERS of CACHE)))
        (HELP "Cache tail pointer is wrong"))
       (CHECKORDER (for X in (fetch NSPBUFFERS of CACHE) bind EMPTY do (COND
                                                                         [EMPTY (COND
                                                                                   ((NEQ (CAR X)
                                                                                         -1)
                                                                                    (HELP "Cache empty elements
                                                                                          not all at end"]
                                                                         ((EQ (CAR X)
                                                                              -1)
                                                                          (SETQ EMPTY T])
```

(\NSRANDOM.WRITEPAGES

```
  [LAMBDA (STREAM FIRSTPAGE# BUFFERS)                            ; Edited 9-Oct-87 15:52 by bvm:
    ;; Write pages method for NS random access file.
    (PROG (SESSION)
          (for BUF inside BUFFERS as P from FIRSTPAGE# do
                                     ;; Flush these pages from cache if they happen to have been prefetched.  Problem is that prefetch
                                     ;; doesn't see what the stream itself has buffered in its pmap buffers, so could have fetched a page
                                     ;; even though there is a local copy, possibly dirty even.
                                                  (\NSRANDOM.FETCH.CACHE STREAM P))
      RETRY
          (COND
             ((NULL (SETQ SESSION (fetch NSFILING.CONNECTION of STREAM)))
                                                                   ; Session lost, e.g., after logout
              (\NSRANDOM.REESTABLISH STREAM)
              (GO RETRY)))
          (LET ((CURRENTEOF (fetch NSFILING.SERVER.LENGTH of STREAM))
                (HANDLE (fetch NSFILING.HANDLE of STREAM))
                (FIRSTBYTE (UNFOLD FIRSTPAGE# BYTESPERPAGE))
                BYTES-TIL-EOF BYTESTOSTORE ATTRS ERROR LASTPAGE)
               [SETQ BYTESTOSTORE (for BUF inside BUFFERS as old LASTPAGE from FIRSTPAGE#
```

```
                                bind (EP _ (fetch (STREAM EPAGE) of STREAM))
                                sum (COND
                                       ((EQ LASTPAGE EP)
                                        (fetch (STREAM EOFFSET) of STREAM))
                                       (T BYTESPERPAGE]
            (COND
               ((EQ BYTESTOSTORE 0)                              ; Nothing to write.  Stupid of pmap to call us.
                (RETURN)))
            [COND
               ((fetch REVALIDATEFLG of STREAM)
                ;; Need to update creationdate, since a SAVEVM etc has occurred since the last write.  Otherwise, it is possible to see a
                ;; change to the file but no change to the creationdate
                (OR (\NSRANDOM.UPDATE.VALIDATION STREAM SESSION HANDLE)
                    (GO RETRY]
            (COND
               ((< (SETQ BYTES-TIL-EOF (- CURRENTEOF FIRSTBYTE))
                   0)                                            ; Writing past end of file?
                (\NSRANDOM.TRUNCATEFILE STREAM (FOLDLO FIRSTBYTE BYTESPERPAGE)
                       (IMOD FIRSTBYTE BYTESPERPAGE))
                (SETQ CURRENTEOF FIRSTBYTE)
                (SETQ BYTES-TIL-EOF 0)))
            [SETQ ERROR (COND
                           [(AND (< BYTES-TIL-EOF BYTESTOSTORE)
                                 (NEQ BYTES-TIL-EOF 0))          ; Range to write overlaps eof.  Filing doesn't like this, so write the
                                                                 ; first part, then the last part
                            (OR (\NSRANDOM.WRITE.SEGMENT SESSION HANDLE BUFFERS FIRSTBYTE BYTES-TIL-EOF)
                                (\NSRANDOM.WRITE.SEGMENT SESSION HANDLE (COND
                                                                          ((NLISTP BUFFERS)
                                                                           BUFFERS)
                                                                          (T (CL:NTHCDR (FOLDLO
                                                                                          BYTES-TIL-EOF

                                                                                           BYTESPERPAGE
                                                                                           )
                                                                                        BUFFERS)))
                                                        CURRENTEOF
                                                        (- BYTESTOSTORE BYTES-TIL-EOF]
                           (T                                    ; Ok to write in one segment
                              (\NSRANDOM.WRITE.SEGMENT SESSION HANDLE BUFFERS FIRSTBYTE BYTESTOSTORE]
            (COND
               (ERROR (\NSRANDOM.HANDLE.ERROR ERROR STREAM SESSION 'REPLACE.BYTES)
                      (GO RETRY)))
            (\NSRANDOM.WROTE.HANDLE SESSION HANDLE)              ; Writing data to file has (potentially) changed its creationdate.
            [COND
               ((< (- CURRENTEOF FIRSTBYTE)
                   BYTESTOSTORE)                                 ; Wrote to eof, so update remote eof
                (replace NSFILING.SERVER.LENGTH of STREAM with (SETQ CURRENTEOF (+ FIRSTBYTE BYTESTOSTORE)))
                (COND
                   ((SETQ ATTRS (ASSOC 'SIZE.IN.BYTES (fetch NSHATTRIBUTES of HANDLE)))
                                                                 ; Update cached info about size of file
                    (CL:SETF (CADR ATTRS)
                           CURRENTEOF]
            (replace NSFILING.LAST.REQUEST of STREAM with LASTPAGE])
```

(\NSRANDOM.WRITE.SEGMENT

```
[LAMBDA (SESSION HANDLE BUFFERS FIRSTBYTE BYTESTOSTORE)          ; Edited  1-Jun-87 16:45 by bvm:
```

;; Write data from BUFFERS, a set of page buffers. FIRSTBYTE is the first byte in file to replace, BYTESTOSTORE the count. If FIRSTBYTE is ;; not on a page boundary, start in the middle of a page.

```
(FILING.CALL SESSION 'REPLACE.BYTES (fetch NSHDATUM of HANDLE)
       (COURIER.CREATE (FILING . BYTE.RANGE)
              FIRSTBYTE _ FIRSTBYTE COUNT _ BYTESTOSTORE)
       [FUNCTION (LAMBDA (BULKSTREAM)
                   ;; What to store as the bulk data
                   (for BUF inside BUFFERS bind (OFFSET _ (IMOD FIRSTBYTE BYTESPERPAGE))
                                                (BYTESLEFT _ BYTESTOSTORE)
                                                CNT
                      do [SETQ BYTESLEFT (COND
                                           ((> (SETQ CNT (- BYTESPERPAGE OFFSET))
                                               BYTESLEFT)
                                            (SETQ CNT BYTESLEFT)
                                            0)
                                           (T (- BYTESLEFT CNT]
                         (\BOUTS BULKSTREAM BUF OFFSET CNT)
                         (SETQ OFFSET 0)
                      repeatuntil (EQ BYTESLEFT 0]
       SESSION
       'RETURNERRORS])
```

(\NSRANDOM.WROTE.HANDLE

```
[LAMBDA (SESSION HANDLE)                                         ; Edited  9-Oct-87 15:52 by bvm:
```

;; Called when we did something (e.g., ReplaceBytes) that would cause the creation date to change. We assume this happens only once per ;; handle, that the file service does not change the date on every write! Since validation depends on creationdate, we have to actually refetch it, not

```
;; just zap it.
[COND
   ((NOT (fetch NSHWASWRITTEN of HANDLE))
    (LET [(NEWINFO (FILING.CALL SESSION 'GET.ATTRIBUTES (fetch NSHDATUM of HANDLE)
                        [CONSTANT (\FILING.ATTRIBUTE.TYPE.SEQUENCE '(CREATED.ON]
                        SESSION
                        'RETURNERRORS]
         (COND
            ((AND NEWINFO (NEQ (CAR NEWINFO)
                               'ERROR))                    ; If error occurred, we don't care, since the handle is then trash.
             (\NSFILING.UPDATE.ATTRIBUTES HANDLE NEWINFO)
             (replace NSHWASWRITTEN of HANDLE with T]
(COND
   ((NOT (fetch NSHWASMODIFIED of HANDLE))                 ; Ditto write date.
    (LET [(ATTR (ASSOC 'MODIFIED.ON (fetch NSHATTRIBUTES of HANDLE]
         [COND
            (ATTR (replace NSHATTRIBUTES of HANDLE with (DREMOVE ATTR (fetch NSHATTRIBUTES of HANDLE]
         (replace NSHWASMODIFIED of HANDLE with T])
```

(\NSRANDOM.SETEOFPTR

```
[LAMBDA (STREAM NBYTES)                                    ; Edited 9-Jun-87 14:03 by bvm:
;; Change open stream length to be NBYTES.  This is our own version of SETEOFPTR, since we have no need to remap the last page.
(LET ((NEWEP (fetch (BYTEPTR PAGE) of NBYTES))
      (NEWEO (fetch (BYTEPTR OFFSET) of NBYTES)))
     (SELECTQ (\NEWLENGTHIS STREAM NEWEP NEWEO)
         (SHORTER (COND
                     ((OVERWRITEABLE STREAM)
                      (FORGETPAGES STREAM (ADD1 NEWEP)
                             (PROG1 (fetch EPAGE of STREAM)   ; \seteof changes EPAGE
                                 (\SETEOF STREAM NEWEP NEWEO)))
                      ;; FORGETPAGES tells PMAP to throw away the extra pages.  The \SETEOF is done first so that an interrupt will
                      ;; not leave STREAM pointing to old and possibly partially overwritten pages.
                      (\NSRANDOM.TRUNCATEFILE STREAM NEWEP NEWEO)
                                                           ; Shorten the real file
                      T)))
         (SAME                                             ; Nothing to do
               T)
         (LONGER (COND
                    ((APPENDABLE STREAM)
                     (\SETEOF STREAM NEWEP NEWEO)
                     T)))
         (SHOULDNT])
```

(\NSRANDOM.TRUNCATEFILE

```
[LAMBDA (STREAM LP LO)                                     ; Edited 9-Oct-87 15:52 by bvm:
;; Resets the length of the file to LP page and LO offset.  Can both shorten and lengthen files.
[PROG (SESSION CURRENTEOF NEWEOF)
  RETRY
       (COND
          ([NOT (= (SETQ CURRENTEOF (fetch NSFILING.SERVER.LENGTH of STREAM))
                   (SETQ NEWEOF (COND
                                   (LP (create BYTEPTR
                                               PAGE _ LP
                                               OFFSET _ LO))
                                   (T (\GETEOFPTR STREAM]     ; There's something to do
           (COND
              ((NULL (SETQ SESSION (fetch NSFILING.CONNECTION of STREAM)))
                                                           ; Session lost, e.g., after logout
               (\NSRANDOM.REESTABLISH STREAM)
               (GO RETRY)))
           (LET ((HANDLE (fetch NSFILING.HANDLE of STREAM))
                 ERROR ATTRS)
                [COND
                   ((fetch REVALIDATEFLG of STREAM)
                    ;; Need to update creationdate, since a SAVEVM etc has occurred since the last write.  Otherwise, it is possible to see a
                    ;; change to the file but no change to the creationdate
                    (OR (\NSRANDOM.UPDATE.VALIDATION STREAM SESSION HANDLE)
                        (GO RETRY]
                ;; Although you might think the right way to shorten a file is to do a ReplaceBytes on the range [newEof,EndOfFile] with zero
                ;; bytes, the server rejects that.  Instead, explicitly change the LENGTH attribute.
                (SETQ ERROR (FILING.CALL SESSION 'CHANGE.ATTRIBUTES (fetch NSHDATUM of HANDLE)
                                 `((SIZE.IN.BYTES ,NEWEOF))
                                 SESSION
                                 'RETURNERRORS))
                (COND
                   (ERROR (\NSRANDOM.HANDLE.ERROR ERROR STREAM SESSION 'CHANGE.ATTRIBUTES)
                          (GO RETRY)))
                (replace NSFILING.SERVER.LENGTH of STREAM with NEWEOF)
                (COND
                   ((SETQ ATTRS (ASSOC 'SIZE.IN.BYTES (fetch NSHATTRIBUTES of HANDLE)))
```

```
                                                                    ; Update cached info about size of file
                        (CL:SETF (CADR ATTRS)
                                 NEWEOF)))
                    (\NSRANDOM.WROTE.HANDLE SESSION HANDLE]
    STREAM])
```

(\NSRANDOM.UPDATE.VALIDATION

```
  [LAMBDA (STREAM SESSION HANDLE)                                   ; Edited  1-Jun-87 16:45 by bvm:
    ;; Called when STREAM's REVALIDATEFLG is true, meaning we need to update its creationdate to ensure that what we are about to write is
    ;; noticeable if we were to boot back to the last savevm.
    (LET* [[NEWATTRS `((CREATED.ON ,(IDATE]
           (ERROR (FILING.CALL SESSION 'CHANGE.ATTRIBUTES (fetch NSHDATUM of HANDLE)
                               NEWATTRS SESSION 'RETURNERRORS]
          (COND
             (ERROR (\NSRANDOM.HANDLE.ERROR ERROR STREAM SESSION 'CHANGE.ATTRIBUTES)
                                                                    ; Return NIL on failure
                    NIL)
             (T (replace REVALIDATEFLG of STREAM with NIL)
                (\NSFILING.UPDATE.ATTRIBUTES HANDLE NEWATTRS])
```

(\NSRANDOM.OPENFILE

```
  [LAMBDA (FILENAME ACCESS RECOG PARAMETERS DEVICE)                 ; Edited  9-Feb-88 15:59 by bvm:
    ;; OPENFILE on the random device.  This only happens when reopening a stream.
    (SETQ FILENAME (if (NOT (type? STREAM FILENAME))
                       then (SHOULDNT "Random OPENFILE called on non-stream")
                     elseif (OPENED FILENAME)
                       then (\ILLEGAL.ARG FILENAME)
                     else                                           ; Reopening a closed stream, such as TEdit might do.
                          (fetch FULLFILENAME of FILENAME)))
    (\NSFILING.OPENFILE FILENAME ACCESS RECOG PARAMETERS (fetch DEVICEINFO of DEVICE])

)

;; error handling

(DEFINEQ

(\NSRANDOM.HANDLE.ERROR
  [LAMBDA (ERROR STREAM SESSION PROCEDURE)                          ; Edited 27-Aug-87 11:30 by bvm:
    ;; Handle error in call to filing random access procedure.  Most interesting one now is session error, which happens when the session times out.
    (SELECTQ (CADR ERROR)
        (SESSION.ERROR
            (LET [(DEVICE (fetch DEVICEINFO of (fetch DEVICE of STREAM]
                 (for S in (fetch OPENFILELST of DEVICE) when (EQ (fetch NSFILING.CONNECTION of S)
                                                                  SESSION)
                    do                                          ; Invalidate all streams on this connection so we're not tempted
                                                                ; to use it again.
                       (replace NSFILING.CONNECTION of S with NIL))
                 (AND (\NSFILING.GET.NEW.SESSION SESSION DEVICE T)
                      (\NSRANDOM.REESTABLISH STREAM))))
        (SPACE.ERROR                                            ; Ran out of space writing the file
                     (\NSRANDOM.PROCEEDABLE.ERROR STREAM 'XCL:FS-RESOURCES-EXCEEDED
                           (LIST :PATHNAME (fetch FULLFILENAME of STREAM))))
        (ACCESS.ERROR                                           ; Grumble.  Can happen if you open an old file for output, or
                                                                ; create a file in a directory to which you have ADD but not
                                                                ; WRITE access.
                      (\NSRANDOM.PROCEEDABLE.ERROR STREAM "Attempt to ~:[read~;write to~] file ~A failed
                            because: ~A.  How shall I proceeed?" (LIST (NEQ PROCEDURE 'RETRIEVE.BYTES)
                                                                       (FULLNAME STREAM)
                                                                       (CADDR ERROR))))
        (TRANSFER.ERROR                                         ; Something went wrong in transit.  let's try it again.
            (PRINTOUT PROMPTWINDOW T "Access to " (FULLNAME STREAM)
                   " failed because: "
                   (CADDR ERROR)
                   "; " "will retry."))
        (COURIER.SIGNAL.ERROR (fetch FSPROTOCOLNAME of SESSION)
              PROCEDURE ERROR])
```

(\NSRANDOM.PROCEEDABLE.ERROR

```
  [LAMBDA (STREAM ERROR ERRORARGS PROCEED-DETAILS)                  ; Edited  5-Feb-88 13:45 by amd
    ;; Enter the debugger because of a problem with STREAM.  ERROR and ERRORARGS are passed to CL:ERROR.  PROCEED-DETAILS, if
    ;; non-NIL, is a format string describing what will happen if you choose the proceed option PROCEED (or OK).  Returns only if CONTINUE was
    ;; selected.
    (CONDITIONS:RESTART-CASE (CL:APPLY (FUNCTION CL:ERROR)
                                       ERROR ERRORARGS)
           (CONDITIONS:CONTINUE NIL :REPORT (CL:LAMBDA (ERR-STRM)
                                              (CL:FORMAT ERR-STRM (OR PROCEED-DETAILS "Try again")
                                                     ERRORARGS)))
           (GIVEUP NIL :REPORT "Abort: close the stream and abort the computation" (\NSRANDOM.DESTROY.STREAM
                                                                                     STREAM)
```

```
                                                        ; Blow away the stream.
                (ERROR!])
```

(\NSRANDOM.REESTABLISH

```
[LAMBDA (STREAM)                                        ; Edited 20-Nov-87 17:08 by bvm:
  (PROG (HANDLE)
    RETRY
        (RETURN (if (NULL (SETQ HANDLE (fetch NSFILING.HANDLE of STREAM)))
                    then                                ; Somebody's already blown away this stream
                         (\NSRANDOM.PROCEEDABLE.ERROR STREAM "Trying to operate on stream after it's
                                 closed: ~S" (LIST STREAM))
                         (GO RETRY)
                  elseif (\NSFILING.GETFILE (fetch DEVICEINFO of (fetch DEVICE of STREAM))
                                (LET ((ID (fetch NSHFILEID of HANDLE)))
                                     (OR (AND ID (LIST 'FILE.ID ID))
                                         (fetch FULLFILENAME of STREAM)))
                                (fetch ACCESS of STREAM)
                                'OLD NIL NIL NIL NIL STREAM)
                  else (\NSRANDOM.PROCEEDABLE.ERROR STREAM "Lost connection to file ~A, can't reestablish"
                                (LIST (fetch FULLFILENAME of STREAM)))
                       (GO RETRY])
```

(\NSRANDOM.STREAM.CHANGED

```
[LAMBDA (OLDSTREAM NEWHANDLE)                           ; Edited  5-Aug-87 16:35 by bvm:
```

;; Called when trying to reestablish OLDSTREAM. NEWHANDLE is a new handle on the file, which shows that the file has changed with respect to
;; OLDSTREAM's handle. Returning from this function will continue by using the new handle.

```
  (\NSRANDOM.PROCEEDABLE.ERROR OLDSTREAM "The file ~A has been modified since you last accessed it.  How
         shall I proceed?" (LIST (FULLNAME OLDSTREAM))
         (COND
            ((DIRTYABLE OLDSTREAM)
             "Continue output to the file, possibly overwriting its more recent contents")
            (T "Continue, reading the new contents of the file")))
  [COND
     ((NEQ (fetch ACCESS of OLDSTREAM)
           'OUTPUT)                                     ; reset eof to correct value
      (LET [(LEN (IMAX (CADR (ASSOC 'SIZE.IN.BYTES (fetch NSHATTRIBUTES of NEWHANDLE]
           (replace EPAGE of OLDSTREAM with (FOLDLO LEN BYTESPERPAGE))
           (replace EOFFSET of OLDSTREAM with (IMOD LEN BYTESPERPAGE]
  (replace NSFILING.PAGE.CACHE of OLDSTREAM with NIL])
```

(\NSRANDOM.DESTROY.STREAM

```
[LAMBDA (STREAM)                                        ; Edited  3-Jun-87 18:58 by bvm:
```

;; Blow away stream in a way that we won't keep dying. CLOSEF will just keep trying to write pages otherwise.

```
  (UNINTERRUPTABLY
      (\RELEASECPAGE STREAM))
  (FORGETPAGES STREAM)                                  ; Discard buffers before closing file, so that CLOSEF doesn't try
                                                        ; to write anything.
  (replace NSFILING.SERVER.LENGTH of STREAM with (\GETEOFPTR STREAM))
                                                        ; Wrong, but it keeps truncatefile from trying to resize the file.
  (CLOSEF STREAM])
```

(\NSRANDOM.SESSION.WATCHER

```
[LAMBDA (DEVICE)                                        ; Edited 10-Jun-87 17:57 by bvm:
```

;; Process that makes sure sessions stay open on DEVICE if they are needed. There are two notions of timeout here: (1) the server has an
;; inactivity timeout; if no courier calls in that time, the session is discarded. (2) we have a timeout for open streams; if no stream activity happens
;; within that time, we are willing to let session die. Our timeout is, in general, greater than the servers; it is chosen to obtain a balance between the
;; expense of opening a new session and reestablishing open streams on it and the load we place on the server by keeping a session open that we
;; aren't actively using.

```
  (LET ((DEVINFO (fetch DEVICEINFO of DEVICE)))
       (replace NSWATCHERPROC of DEVINFO with (THIS.PROCESS))   ; Redundant ordinarily (ensure.watcher does this itself to avoid
                                                                ; races), but important to redo it after HARDRESET.
       (RESETSAVE NIL (LIST [FUNCTION (LAMBDA (DEVINFO)         ; Remove this pointer when process goes away
                                      (replace NSWATCHERPROC of DEVINFO with NIL]
                            DEVINFO))
       (do (LET (WRITING? SESSION TIMEOUT CONTINUANCE BASICTIMER)
                                                                ; See if any random access files are open
                (COND
                   ([NULL (SETQ SESSION (CAR (fetch NSCONNECTIONS of DEVINFO]
                                                                ; No live sessions, so nothing to watch
                    (RETURN))
                   ([NOT (for S in (fetch (FDEV OPENFILELST) of DEVICE) when (NEQ (fetch (STREAM DEVICE)
                                                                                        of S)
                                                                                  DEVICE)
                             do                                 ; Stream is open on random device
                                (SETQ $$VAL T)
                                (COND
                                   ((DIRTYABLE S)
                                    (SETQ WRITING? T]           ; No randaccess files are open, so nothing to watch
                    (RETURN))
                   ([EQ 0 (SETQ TIMEOUT (COND
```

```
                                        ((NLISTP *NSFILING-SESSION-TIMEOUT*)
                                         *NSFILING-SESSION-TIMEOUT*)
                                        (WRITING? (CDR *NSFILING-SESSION-TIMEOUT*))
                                        (T (CAR *NSFILING-SESSION-TIMEOUT*]
                                                                    ; timeout is zero (i.e., timeout immediately), so don't need to stick
                                                                    ; around.
                         (RETURN))
                        ([NOT (\SECONDSCLOCKGREATERP (SETQ BASICTIMER (fetch FSSESSIONTIMER of SESSION))
                                     (SETQ CONTINUANCE (fetch FSCONTINUANCE of SESSION]
                                                                    ; Ho hum, we have lots of time before we need to worry about
                                                                    ; keeping session alive.

                        )
                        ((AND TIMEOUT (\SECONDSCLOCKGREATERP (LET ((REALTIMER (fetch FSLASTREALACTIVITYTIMER
                                                                                 of SESSION)))
                                                               (COND
                                                                  ((AND REALTIMER (NOT (fetch FSREALACTIVITY
                                                                                          of SESSION)))
                                                                                    ; nothing's happened since the last CONTINUE
                                                                     REALTIMER)
                                                                  (T BASICTIMER)))
                                                    TIMEOUT))       ; Real timeout has passed, so give up
                         (RETURN))
                        ((NOT (FILING.CALL SESSION 'CONTINUE SESSION 'NOERROR))
                                                                    ; Failed to keep the session alive, go away
                         (RETURN)))
                      (BLOCK (TIMES 1000 (IMAX (- CONTINUANCE (- (\DAYTIME0 (\CREATECELL \FIXP))
                                                                 BASICTIMER))
                                               0)))                 ; Dismiss until the time we next worry about session going away.

                     ])
```

(\NSRANDOM.ENSURE.WATCHER

```
  [LAMBDA (DEVICE)                                                  ; Edited 2-Jun-87 15:33 by bvm:
```

;; Create a watcher process for this device, if one does not already exist, to make sure that sessions stay open.

```
    (LET ((DEVINFO (fetch DEVICEINFO of DEVICE)))
         (OR (fetch NSWATCHERPROC of DEVINFO)
             (replace NSWATCHERPROC of DEVINFO with (ADD.PROCESS (LIST (FUNCTION \NSRANDOM.SESSION.WATCHER)
                                                                       DEVICE)
                                                                'RESTARTABLE
                                                                'HARDRESET
                                                                'NAME
                                                                (CONCAT (fetch NSFILINGNAME of DEVINFO)
                                                                        " Watcher")
                                                                'AFTEREXIT
                                                                'DELETE])

)

;; Cleaning up directories

(DEFINEQ

(GC-FILING-DIRECTORY
  [LAMBDA (DIRNAME CONFIRM?)                                        ; Edited 5-Aug-87 15:20 by bvm:
```

;; Device method for file enumeration. Return a generator that enumerates files matching PATTERN. DESIREDPROPS is set of attributes caller ;; may ask for. If OPTIONS includes RESETLST, caller promises to be wrapped in a RESETLST that we can use to kill an aborted bulk listing.

```
    (if (OR (NULL DIRNAME)
            (NEQ (CHCON1 DIRNAME)
                 (CHARCODE "{")))
        then                                                        ; add defaults
          (SETQ DIRNAME (\ADD.CONNECTED.DIR DIRNAME)))
    (PROG ((DEVICE (\GETDEVICEFROMNAME DIRNAME))
           (PARSE (\NSFILING.PARSE.FILENAME DIRNAME T))
           (NDELETED 0)
           CANDIDATES HOST DIRINDEX TOPID)
          (COND
             ((NEQ (fetch OPENFILE of DEVICE)
                   (FUNCTION \NSFILING.OPENFILE))
              (RETURN "Not an NS File Server"))
             ((NOT (fetch NSDIRECTORYP of PARSE))
              (RETURN "Not a directory name"))
             ((OR (NLISTP (SETQ CANDIDATES (\NSGC.COLLECT.DIRECTORIES DEVICE (fetch NSDIRECTORIES of PARSE)
                                                                      T)))
                  (EQ (CAR CANDIDATES)
                      'ERROR))                                      ; Some sort of failure
              (RETURN CANDIDATES)))
          (SETQ TOPID (pop CANDIDATES))
          (COND
             ((NULL CANDIDATES)
              (RETURN "No empty directories")))
```

;; Now have list of file id's that are directories with no children.

```
          (PRINTOUT T "{" [SETQ HOST (fetch FSNAMESTRING of (CAR (fetch NSCONNECTIONS of (fetch DEVICEINFO
```

```
                                                                                    of DEVICE]
          "}" T)
    (SETQ DIRINDEX (+ 3 (NCHARS HOST)))                    ; Index of where directory name will start in full names.
    (for ID in CANDIDATES
       do (while (AND (SETQ ID (\NSFILING.GETFILE
                                DEVICE
                                `(FILE.ID ,ID)
                                'NONE NIL 'HANDLE
                                [FUNCTION (LAMBDA (SESSION HANDLE)
                                                  (COND
                                                     ((EQ (fetch NSHBUSYCOUNT of HANDLE)
                                                          0)      ; Directory not in use, ok to delete.  Ordinarily nobody holds on to
                                                                  ; directories, so this may be superfluous today
                                                      (for PAIR in (FILING.CALL SESSION 'GET.ATTRIBUTES
                                                                                (fetch NSHDATUM of HANDLE)
                                                                                (\FILING.ATTRIBUTE.TYPE.SEQUENCE
                                                                                 '(NUMBER.OF.CHILDREN PARENT.ID))
                                                                                SESSION)
                                                         bind PARENT ERROR
                                                         do (SELECTQ (CAR PAIR)
                                                                     (NUMBER.OF.CHILDREN
                                                                         (COND
                                                                            ((NEQ (CADR PAIR)
                                                                                  0)
                                                                  ; Has children now, skip it.  Note that this could be true for any
                                                                  ; directory collected above, because we didn't obtain handles
                                                                  ; then.
                                                                             (RETURN NIL))))
                                                                     (PARENT.ID (SETQ PARENT (CADR PAIR)))
                                                                     (SHOULDNT))
                                                         finally     ; Ready to try deleting it.
                                                                (PRINTOUT T (SUBSTRING (\NSFILING.FULLNAME SESSION
                                                                                                           HANDLE)
                                                                                       DIRINDEX)
                                                                      %,)
                                                                (COND
                                                                   ((AND CONFIRM?
                                                                         (NEQ 'Y (ASKUSER NIL NIL "delete? "
                                                                                          '((Y "es ")
                                                                                            (N "o "))
                                                                                          T)))
                                                                  ; disconfirmed

                                                                    )
                                                                   ((SETQ ERROR (FILING.CALL SESSION 'DELETE
                                                                                             (fetch NSHDATUM of HANDLE)
                                                                                             SESSION
                                                                                             'RETURNERRORS))
                                                                    (COND
                                                                       ((EQ (CADDR ERROR)
                                                                            'TokenInvalid)
                                                                  ; sigh, could get this if the ASKUSER took a long time.  Go
                                                                  ; around again
                                                                        (PRINTOUT T "connection lost" T)
                                                                        (RETURN ID)))
                                                                    (PRINTOUT T (CADDR ERROR)))
                                                                   (T
                                                                  ; success
                                                                      (PRINTOUT T "deleted." T)
                                                                      (add NDELETED 1)
                                                                      (replace FSCACHEDHANDLES of SESSION
                                                                         with (DREMOVE HANDLE (fetch FSCACHEDHANDLES
                                                                                                     of SESSION)))
                                                                  ; return parent id for another go around in case deleting this
                                                                  ; directory emptied the parent.
                                                                      (RETURN PARENT)))
                                                                (TERPRI T)
                                                                (RETURN NIL]
                                                     T))
                                (NOT (EQUAL ID TOPID)))
             do                                                ; Keep trying to delete dirs until we get back to the root.
          ))
    (RETURN (CONCAT NDELETED " directories deleted"])
```

(\NSGC.COLLECT.DIRECTORIES

```
  [LAMBDA (DEVICE DIRPATH NOCHILDREN)                      ; Edited  5-Aug-87 15:20 by bvm:
```

;; Return a list of directory id's below DIRPATH, with the root directory's id consed on the front. If NOCHILDREN is true, only directories with zero children are included.

```
    (RESETLST                                              ; Need RESETLST for \getfilingconnection
        [PROG ([SCOPELIST `((DEPTH 65535)
                            (FILTER (AND ((= ((IS.DIRECTORY T)
                                              BOOLEAN))
                                         ,@(AND NOCHILDREN '((= ((NUMBER.OF.CHILDREN 0)
                                                                 CARDINAL]
               (SESSION (\GETFILINGCONNECTION DEVICE))
```

```
               BULKSTREAM HANDLE GENERATOR)
              (COND
                 ((NULL SESSION)
                  (RETURN NIL))
                 ((NULL (SETQ HANDLE (\NSFILING.CONNECT SESSION DIRPATH T)))
                  (RETURN "No such directory")))
          RETRY
              (SETQ BULKSTREAM (FILING.CALL SESSION 'LIST (fetch NSHDATUM of HANDLE)
                                       [if NOCHILDREN
                                           then [CONSTANT (\FILING.ATTRIBUTE.TYPE.SEQUENCE '(FILE.ID IS.DIRECTORY

                                                                                                NUMBER.OF.CHILDREN
                                                                                                   ]
                                         else (CONSTANT (\FILING.ATTRIBUTE.TYPE.SEQUENCE '(FILE.ID IS.DIRECTORY]
                                      SCOPELIST NIL SESSION 'RETURNERRORS 'KEEPSTREAM))
              (COND
                 ([AND (LISTP BULKSTREAM)
                       (CDR SCOPELIST)
                       (EQUAL BULKSTREAM '(ERROR SCOPE.VALUE.ERROR Unimplemented FILTER]
                                                                  ; Would be nice to have a filter on IS.DIRECTORY and
                                                                  ; NUMBER.OF.CHILDREN, but servers don't implement that.
                  [SETQ SCOPELIST '((DEPTH 65535]
                  (GO RETRY)))
              (COND
                 ((NOT (STREAMP BULKSTREAM))
                  (RETURN BULKSTREAM)))
              (RESETSAVE NIL (LIST (FUNCTION \NSFILING.CLOSE.BULKSTREAM)
                                   SESSION BULKSTREAM))
              (SETQ GENERATOR (BULKDATA.GENERATOR BULKSTREAM (fetch FSPROTOCOLNAME of SESSION)
                                    'ATTRIBUTE.SEQUENCE))
              (RETURN (CONS (fetch NSHFILEID of HANDLE)
                            (bind ID INFO eachtime (SETQ ID NIL) while (SETQ INFO (BULKDATA.GENERATE.NEXT
                                                                                    GENERATOR))
                               when (for PAIR in INFO always (SELECTQ (CAR PAIR)
                                                                  (FILE.ID (SETQ ID (CADR PAIR)))
                                                                  (IS.DIRECTORY (CADR PAIR))
                                                                  (NUMBER.OF.CHILDREN
                                                                      (EQ 0 (CADR PAIR)))
                                                                  NIL))
                               collect ID])])

)

;; Deserialize (special for NSMAIL)

(DEFINEQ

(\NSFILING.DESERIALIZE
  [LAMBDA (FILENAME SERIALSTREAM DEVICE)                          ; Edited 8-Dec-87 13:05 by bvm:
    (RESETLST
        [LET ((PARSE (\NSFILING.PARSE.FILENAME FILENAME))
              DIRHANDLE HANDLE SESSION VERSION NAME)
             (COND
                ((NULL PARSE)                                     ; Bad name
                 (CL:ERROR 'XCL:INVALID-PATHNAME :PATHNAME FILENAME))
                ((NULL (SETQ SESSION (\GETFILINGCONNECTION DEVICE)))
                 (CL:ERROR 'XCL:FILE-NOT-FOUND :PATHNAME FILENAME))
                ((NULL (SETQ DIRHANDLE (\NSFILING.CONNECT SESSION (fetch NSDIRECTORIES of PARSE)
                                              T T)))              ; Couldn't get handle on destination
                 (CL:ERROR 'XCL:FILE-WONT-OPEN :PATHNAME FILENAME))
                [(AND (LISTP (SETQ HANDLE (\NSFILING.DESERIALIZE1
                                           SESSION DIRHANDLE
                                           `[,@[AND (SETQ NAME (fetch NSROOTNAME of PARSE))
                                                    `((NAME ,(\NSFILING.REMOVEQUOTES NAME]
                                             ,@(AND (SETQ VERSION (fetch NSVERSION of PARSE))
                                                    `((VERSION ,(MKATOM VERSION]
                                           SERIALSTREAM)))
                      (NEQ (CAR HANDLE)
                           'ERROR))                               ; Success
                 (\NSFILING.FULLNAME SESSION (\NSFILING.ADD.TO.CACHE SESSION (create FILINGHANDLE
                                                                                    NSHDATUM _ HANDLE]
                (T                                                ; Failure
                   (COURIER.SIGNAL.ERROR (fetch FSPROTOCOLNAME of SESSION)
                          'DESERIALIZE HANDLE])])

(\NSFILING.DESERIALIZE1
  [LAMBDA (SERIALSESSION DIRHANDLE NEWATTRS SERIALSTREAM CLOSEFN)
                                                                  ; Edited 9-Dec-87 18:27 by bvm:
    ;; Perform the DESERIALIZE call on SESSION, handle of parent directory, attributes to change, and the source of the serialized file.  The awful
    ;; contorted structure is so we don't tie up the session while the transfer is in progress.
    (LET [(BULKSTREAM (FILING.CALL SERIALSESSION 'DESERIALIZE (fetch NSHDATUM of DIRHANDLE)
                           NEWATTRS NIL NIL SERIALSESSION 'RETURNERRORS 'KEEPSTREAM]
         (CL:UNWIND-PROTECT
             (LET (EXPLICIT-RESULT BULKRESULT)
                  (RELEASE.MONITORLOCK (fetch FSSESSIONLOCK of SERIALSESSION))
```

```
                                                        ; Don't let this serial transfer tie up the session forever.
            (SETQ EXPLICIT-RESULT (if (TYPENAMEP SERIALSTREAM 'STREAM)
                                      then              ; a stream containing the serialized data
                                           (COPYBYTES SERIALSTREAM BULKSTREAM)
                                                        ; Normally want to return NIL from here so we see the real
                                                        ; courier results.
                                           (AND CLOSEFN (CL:FUNCALL CLOSEFN BULKSTREAM))
                                      else              ; A function to store the file.
                                          (CL:FUNCALL SERIALSTREAM BULKSTREAM)))
            [SETQ BULKRESULT (\BULK.DATA.CLOSE BULKSTREAM (AND (LISTP EXPLICIT-RESULT)
                                                               (EQ (CAR EXPLICIT-RESULT)
                                                                   'ERROR]
            (OBTAIN.MONITORLOCK (fetch FSSESSIONLOCK of SERIALSESSION))
            (OR EXPLICIT-RESULT BULKRESULT))
        ;; Cleanups: Abort bulk data if there's a problem, release bulk stream
        (\BULK.DATA.CLOSE BULKSTREAM T)
        (\NSFILING.RELEASE.BULKSTREAM SERIALSESSION BULKSTREAM))])

)

(DEFINEQ

(\NSFILING.INIT
  [LAMBDA NIL                                           ; Edited 15-May-87 17:15 by bvm:
    (\DEFINEDEVICE NIL (create FDEV
                               DEVICENAME _ 'NSFILING
                               HOSTNAMEP _ (FUNCTION \NSFILING.HOSTNAMEP)
                               EVENTFN _ (FUNCTION NILL)))
    (DEFPRINT 'FILINGSESSION (FUNCTION \FILINGSESSION.DEFPRINT))
    (DEFPRINT 'FILINGHANDLE (FUNCTION \FILINGHANDLE.DEFPRINT])

)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(\NSFILING.INIT)
)

(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS

(ADDTOVAR NLAMA )

(ADDTOVAR NLAML )

(ADDTOVAR LAMA FILING.CALL)
)
```

FUNCTION INDEX

VARIABLE INDEX

CONSTANT INDEX

RECORD INDEX

PROPERTY INDEX

COURIERPROGRAM INDEX

MACRO INDEX