```
(RPAQQ FILEIOCOMS
       [(PROP (FILETYPE MAKEFILE-ENVIRONMENT)
              FILEIO)

        ;; Device independent IO.  This file is used by VAX

        (COMS ;; STREAM, FDEV declarations
              (DECLARE%: FIRST DOCOPY

                     ;; The microcode relies on STREAM being of a particular type, viz.  the first type declared in the initial loadup (after
                     ;; VMEMPAGEP)

                     (INITRECORDS STREAM))
              (SYSRECORDS STREAM)
              (DECLARE%: DONTCOPY (EXPORT (RECORDS STREAM)
                                          (MACROS STREAMOP)
                                          (CONSTANTS AppendBit NoBits ReadBit WriteBit (OutputBits (LOGOR
                                                                                                          AppendBit

                                                                                                          WriteBit
                                                                                                          ))
                                                     (BothBits (LOGOR ReadBit OutputBits))
                                                     \NORUNCODE)
                                          (MACROS TestMasked APPENDABLE APPENDONLY DIRTYABLE OPENED OVERWRITEABLE
                                                  READABLE READONLY WRITEABLE)
                                          (MACROS \RUNCODED)
                                          (CONSTANTS * EOLCONVENTIONS)))
              (FNS STREAMPROP GETSTREAMPROP PUTSTREAMPROP STREAMP)
              [COMS                                              ; make streams print pretty
                    (FNS \DEFPRINT.BY.NAME \STREAM.DEFPRINT \FDEV.DEFPRINT)
                    (DECLARE%: DONTEVAL@LOAD DOCOPY (P (DEFPRINT 'STREAM (FUNCTION \STREAM.DEFPRINT))
                                                       (DEFPRINT 'FDEV (FUNCTION \FDEV.DEFPRINT]
              (COMS                                              ; Needed because of STREAM initialization
                    (INITVARS (FILELINELENGTH 102)
                              (\STREAM.DEFAULT.MAXBUFFERS 3)))
              (FNS \GETACCESS \SETACCESS)
              (DECLARE%: DONTCOPY (EXPORT (MACROS FDEVOP \RECOGNIZE-HACK)
                                          (RECORDS FDEV FILEGENOBJ)))
              (INITRECORDS FDEV)
              (SYSRECORDS FDEV))
        [COMS                                                    ; EXTERNALFORMAT declaration and related functions
              (DECLARE%: DOEVAL@COMPILE DONTCOPY (EXPORT (RECORDS EXTERNALFORMAT)))
              (INITRECORDS EXTERNALFORMAT)
              (SYSRECORDS EXTERNALFORMAT)
              (FNS \INSTALL.EXTERNALFORMAT \REMOVE.EXTERNALFORMAT \GET.EXTERNALFORMAT.FROM.NAME \EXTERNALFORMAT)
              (INITVARS (*SUPPORTED-EXTERNALFORMATS* '(:XCCS :DEFAULT))
                        (*EXTERNALFORMAT-NAME-TO-EXTERNALFORMAT*))
              (GLOBALVARS *SUPPORTED-EXTERNALFORMATS* *EXTERNALFORMAT-NAME-TO-EXTERNALFORMAT*)
              (FNS \CREATE.JIS.EXTERNALFORMAT \CREATE.SHIFTJIS.EXTERNALFORMAT \CREATE.EUC.EXTERNALFORMAT
                   \CREATE.THROUGH.EXTERNALFORMAT)
              (DECLARE%: DONTEVAL@LOAD DOCOPY (P (\CREATE.JIS.EXTERNALFORMAT)
                                                 (\CREATE.SHIFTJIS.EXTERNALFORMAT)
                                                 (\CREATE.EUC.EXTERNALFORMAT)
                                                 (\CREATE.THROUGH.EXTERNALFORMAT]
        (COMS                                                    ; Device operations
              (FNS \DEFINEDEVICE \GETDEVICEFROMNAME \GETDEVICEFROMHOSTNAME \REMOVEDEVICE \REMOVEDEVICE.NAMES)
              (FNS \CLOSEFILE \DELETEFILE \DEVICEEVENT \GENERATEFILES \GENERATENEXTFILE \GENERATEFILEINFO
                   \GETFILENAME \GENERIC.READCCODE \GENERIC.OUTFILEP \OPENFILE \DO.PARAMS.AT.OPEN \RENAMEFILE
                   \REVALIDATEFILE \PAGED.REVALIDATEFILELST \PAGED.REVALIDATEFILES \PAGED.REVALIDATEFILE
                   \BUFFERED.REVALIDATEFILE \BUFFERED.REVALIDATEFILELST \PRINT-REVALIDATION-RESULT \TRUNCATEFILE
                   \FILE-CONFLICT)
              (COMS                                              ; Generic enumerator
                    (FNS \GENERATENOFILES \NULLFILEGENERATOR \NOFILESNEXTFILEFN \NOFILESINFOFN)
                    (DECLARE%: DONTCOPY (RECORDS NOFILEGENSTATE)))
              (FNS \FILE.NOT.OPEN \FILE.WONT.OPEN \ILLEGAL.DEVICEOP \IS.NOT.RANDACCESSP \STREAM.NOT.OPEN)
              (ADDVARS (\FILEDEVICES)
                       (\FILEDEVICENAMES)
                       (\DEVICENAMETODEVICE))
              (COMS                                              ; Device instances
                    (FNS \FDEVINSTANCE)
```

```
                          (MACROS \INHERITFDEVOP.D \INHERITFDEVOP.S))
                  (INITVARS (LOGINHOST/DIR '{DSK})
                      (\CONNECTED.DIRECTORY '{DSK}))
                  (GLOBALVARS LOGINHOST/DIR \CONNECTED.DIRECTORY \FILEDEVICES \FILEDEVICENAMES \DEVICENAMETODEVICE))
          (COMS                                                  ; Directory defaulting
                  (FNS CNDIR DIRECTORYNAME DIRECTORYNAMEP HOSTNAMEP \ADD.CONNECTED.DIR))
          [COMS                                                  ; Binary I/O Public functions
                  (FNS \BACKFILEPTR \BACKPEEKBIN \BACKBIN BIN \BIN \BINS BOUT \BOUT \BOUTS COPYBYTES COPYCHARS
                       COPYFILE \COPYOPENFILE \INFER.FILE.TYPE EOFP FORCEOUTPUT \FLUSH.OPEN.STREAMS CHARSET
                       ACCESS-CHARSET GETEOFPTR GETFILEINFO \TYPE.FROM.FILETYPE \FILETYPE.FROM.TYPE GETFILEPTR
                       SETFILEINFO SETFILEPTR BOUT16 BIN16)
                  (PROP (DOPCODE)
                       BOUT)
                                                                 ; Generic functions
                  (FNS \GENERIC.BINS \GENERIC.BOUTS \GENERIC.RENAMEFILE \GENERIC.OPENP \GENERIC.READP
                       \GENERIC.CHARSET)
                  (FNS \MAP-OPEN-STREAMS)
                  (VARS FILING.TYPES)
                  (GLOBALVARS FILING.TYPES)
                  (DECLARE%: EVAL@COMPILE DONTCOPY (EXPORT (MACROS \OUTCHAR \DEVICE-OPEN-STREAMS \CONVERT-PATHNAME)
                                                        (OPTIMIZERS ACCESS-CHARSET)))
                  (DECLARE%: DONTEVAL@LOAD DOCOPY (P (MAPC '((FORCEOUTPUT FLUSHOUTPUT)
                                                            (FORCEOUTPUT FLUSHMAP)
                                                            (\GENERIC.BINS \NONPAGEDBINS)
                                                            (\GENERIC.BOUTS \NONPAGEDBOUTS))
                                                      (FUNCTION (LAMBDA (PAIR)
                                                                        (PUTD (CADR PAIR)
                                                                              (GETD (CAR PAIR))
                                                                        T]
          (COMS                                                  ; Internal functions
                  (FNS \EOF.ACTION \EOSERROR \GETEOFPTR \INCFILEPTR \PEEKBIN \SETCLOSEDFILELENGTH \SETEOFPTR
                       \SETFILEPTR)
                  (FNS \FIXPOUT \FIXPIN)
                  (DECLARE%: DONTCOPY (EXPORT (MACROS \DECFILEPTR \GETFILEPTR \SIGNEDWIN \SIGNEDWOUT \WIN \WOUT
                                                     \BINS \BOUTS \EOFP SIZE.FROM.LENGTH)
                                          (CONSTANTS BitsPerByte (ByteOffsetSize (SELECTQ (SYSTEMTYPE)
                                                                                         (VAX 10)
                                                                                         9))
                                                     WordsPerPage)
                                          [CONSTANTS (\MAXFILEPTR (SUB1 (LLSH 1 30]
                                                     (RECORDS BYTEPTR))
                                          (CONSTANTS MaxChar)))
          (COMS                                                  ; Buffered IO
                  (FNS \BUFFERED.BIN \BUFFERED.PEEKBIN \BUFFERED.BOUT \BUFFERED.BINS \BUFFERED.BOUTS
                       \BUFFERED.COPYBYTES))
          [COMS                                                  ; NULL device
                  (FNS \NULLDEVICE \NULL.OPENFILE)
                  (DECLARE%: DONTEVAL@LOAD DOCOPY (P (\NULLDEVICE]
          (LOCALVARS . T)
          (DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILERVARS (ADDVARS (NLAMA)
                                                                               (NLAML)
                                                                               (LAMA \IS.NOT.RANDACCESSP
                                                                                     \ILLEGAL.DEVICEOP STREAMPROP]
)

(PUTPROPS FILEIO FILETYPE :BCOMPL)

(PUTPROPS FILEIO MAKEFILE-ENVIRONMENT (:PACKAGE "INTERLISP" :READTABLE "INTERLISP" :BASE 10))

;; Device independent IO.  This file is used by VAX
;; STREAM, FDEV declarations

(DECLARE%: FIRST DOCOPY

(/DECLAREDATATYPE 'STREAM
        '(WORD WORD FLAG (BITS 3)
               POINTER BYTE BYTE WORD WORD WORD FLAG FLAG FLAG FLAG POINTER FLAG FLAG FLAG FLAG POINTER FLAG
               (BITS 2)
               FLAG POINTER POINTER POINTER WORD WORD POINTER POINTER POINTER POINTER POINTER WORD WORD WORD WORD
               POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER WORD WORD POINTER)

        ;; ---field descriptor list elided by lister---

        '56)
)

(ADDTOVAR SYSTEMRECLST
        (DATATYPE STREAM ((COFFSET WORD)
                          (CBUFSIZE WORD)
                          (PEEKEDCHARP FLAG)
                          (ACCESSBITS BITS 3)
                          (CBUFPTR POINTER)
                          (BYTESIZE BYTE)
                          (CHARSET BYTE)
                          (PEEKEDCHAR WORD)
                          (CHARPOSITION WORD)
                          (CBUFMAXSIZE WORD)
```

```
                                    (NONDEFAULTDATEFLG FLAG)
                                    (REVALIDATEFLG FLAG)
                                    (MULTIBUFFERHINT FLAG)
                                    (USERCLOSEABLE FLAG)
                                    (FULLFILENAME POINTER)
                                    (BINABLE FLAG)
                                    (BOUTABLE FLAG)
                                    (EXTENDABLE FLAG)
                                    (CBUFDIRTY FLAG)
                                    (DEVICE POINTER)
                                    (USERVISIBLE FLAG)
                                    (EOLCONVENTION BITS 2)
                                    (NOTXCCS FLAG)
                                    (VALIDATION POINTER)
                                    (CPAGE POINTER)
                                    (EPAGE POINTER)
                                    (EOFFSET WORD)
                                    (LINELENGTH WORD)
                                    (F1 POINTER)
                                    (F2 POINTER)
                                    (F3 POINTER)
                                    (F4 POINTER)
                                    (F5 POINTER)
                                    (FW6 WORD)
                                    (FW7 WORD)
                                    (FW8 WORD)
                                    (FW9 WORD)
                                    (F10 POINTER)
                                    (STRMBINFN POINTER)
                                    (STRMBOUTFN POINTER)
                                    (OUTCHARFN POINTER)
                                    (ENDOFSTREAMOP POINTER)
                                    (OTHERPROPS POINTER)
                                    (IMAGEOPS POINTER)
                                    (IMAGEDATA POINTER)
                                    (BUFFS POINTER)
                                    (MAXBUFFERS WORD)
                                    (LASTCCODE WORD)
                                    (EXTRASTREAMOP POINTER))))

(DECLARE%: DONTCOPY

;; FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

(DATATYPE STREAM (
```
;; First 8 words are fixed for BIN, BOUT opcodes. Used to require length of whole datatype be multiple of 4, but Dolphin dead
;; now.

```
      (COFFSET WORD)                                    ; Offset in CPPTR of next bin or bout
      (CBUFSIZE WORD)                                   ; Offset past last byte in that buffer
      (PEEKEDCHARP FLAG)                                ; if true, PEEKEDCHAR contains value of recent call to
                                                        ; unread-char
      (ACCESSBITS BITS 3)                              ; What kind of access file is open for (read, write, append)
      (CBUFPTR POINTER)                                ; Pointer to current buffer
      (BYTESIZE BYTE)                                   ; Byte size of stream, always 8 for now
      (CHARSET BYTE)                                    ; the current character set for this stream.  If 255, stream is not
                                                        ; runcoded, so read-char consumes two bytes every time
      (PEEKEDCHAR WORD)                                ; value of unread-char call
      (CHARPOSITION WORD)                              ; Used by POSITION etc.
      (CBUFMAXSIZE WORD)                              ; on output, the size of the physical buffer--can't extend beyond
                                                        ; this
```

;; -------- Above fields (8 words) potentially known to microcode. --------

```
      (NONDEFAULTDATEFLG FLAG)
      (REVALIDATEFLG FLAG)
      (MULTIBUFFERHINT FLAG)                           ; True if stream likes to read and write more than one buffer at a
                                                        ; time
      (USERCLOSEABLE FLAG)                             ; Can be closed by CLOSEF;  NIL for terminal, dribble...
      (FULLFILENAME POINTER)                           ; Name by which file is known to user
      (BINABLE FLAG)                                    ; BIN punts unless this bit on
      (BOUTABLE FLAG)                                   ; BOUT punts unless this bit on
      (EXTENDABLE FLAG)                                ; BOUT punts when COFFSET ge CBUFFSIZE unless this bit
                                                        ; set and COFFSET lt 512
      (CBUFDIRTY FLAG)                                 ; true if BOUT has sullied the current buffer
      (DEVICE POINTER)                                 ; FDEV of this guy
      (USERVISIBLE FLAG)                               ; Listed by OPENP;  NIL for terminal, dribble ...
      (EOLCONVENTION BITS 2)                          ; End-of-line convention
      (NOTXCCS FLAG)                                    ; True if the character encoding format is not XCCS.
      (VALIDATION POINTER)                             ; A number somehow identifying file, used to determine if file has
                                                        ; changed in our absence
      (CPAGE POINTER)                                  ; CPAGE,,COFFSET constitutes current file pointer for most
                                                        ; randaccess streams
      (EPAGE POINTER)
      (EOFFSET WORD)                                   ; Page, byte offset of eof
      (LINELENGTH WORD)                               ; LINELENGTH of stream, or -1 for no line length
```

```
                        ;; ----Following are device-specific fields----

                        ;; Available for device-specific uses, NOT for application use.

                        (F1 POINTER)
                        (F2 POINTER)
                        (F3 POINTER)
                        (F4 POINTER)
                        (F5 POINTER)
                        (FW6 WORD)
                        (FW7 WORD)
                        (FW8 WORD)
                        (FW9 WORD)
                        (F10 POINTER)

                        ;; ----Following only filled in for open streams----

                        (STRMBINFN POINTER)              ; Either the BIN fn from the FDEV, or a trap
                        (STRMBOUTFN POINTER)             ; Either the BIN fn from the FDEV, or a trap
                        (OUTCHARFN POINTER)              ; Called by \OUTCHAR, the normal character printer.
                        (ENDOFSTREAMOP POINTER)          ; Called if EOF and we try to read.
                        (OTHERPROPS POINTER)             ; PROP LIST for holding other info.
                        (IMAGEOPS POINTER)               ; Image operations vector
                        (IMAGEDATA POINTER)              ; Image instance variables--format depends on IMAGEOPS
                                                         ; value
                        (BUFFS POINTER)                  ; Buffer chain for pmapped streams
                        (MAXBUFFERS WORD)                ; Max # of buffers the system will allocate.
                        (LASTCCODE WORD)                 ; After READ, RATOM, etc, the charcode that will be returned (as
                                                         ; a character) by LASTC.  If there is none, this field is 65535.
                        (EXTRASTREAMOP POINTER)          ; For use of applications programs, not devices

                        )
        (BLOCKRECORD STREAM ((NIL 2 WORD)
                             (UCODEFLAGS1 BITS 1)

                        ;; respecification of access bits:

                             (RANDOMWRITEABLE FLAG)      ; File open for output (access = OUTPUT or BOTH)
                             (APPENDABLE FLAG)           ; File open for append (OUTPUT or APPEND or BOTH)
                             (READABLE FLAG)             ; File open for read (READ or BOTH)
                             (NIL POINTER)))
        (BLOCKRECORD STREAM ((NIL 4 WORD)
                             (NIL BITS 14)

                        ;; JIS character encoding format specific, overrides CHARSET field.

                             (IN.KANJIIN FLAG)           ; True if input stream is in Kanji-in mode.
                             (OUT.KANJIIN FLAG)          ; True if output stream is in Kanji-in mode.

                        ))
        [ACCESSFNS STREAM ((ACCESS \GETACCESS \SETACCESS)
                           (FULLNAME (OR (fetch (STREAM FULLFILENAME) of DATUM)
                                         DATUM))
                           (NAMEDP (AND (fetch (STREAM FULLFILENAME) of DATUM)
                                        T]
        [ACCESSFNS STREAM (EXTERNALFORMAT (LISTGET (ffetch (STREAM OTHERPROPS) of DATUM)
                                                   'EXTERNALFORMAT)
                              (LET ((PROPS (ffetch (STREAM OTHERPROPS) of DATUM)))
                                   (freplace (STREAM NOTXCCS) of DATUM with T)
                                   [COND
                                      (PROPS (LISTPUT PROPS 'EXTERNALFORMAT NEWVALUE))
                                      (T (freplace (STREAM OTHERPROPS) of DATUM with (LIST 'EXTERNALFORMAT
                                                                                          NEWVALUE]
                                   (freplace (STREAM OUTCHARFN) of DATUM with (ffetch (EXTERNALFORMAT
                                                                                         FILEOUTCHARFN)
                                                                                  of NEWVALUE))
                                   (AND (ffetch (EXTERNALFORMAT EOLVALID) of NEWVALUE)
                                        (freplace (STREAM EOLCONVENTION) of DATUM with (ffetch (EXTERNALFORMAT
                                                                                                 EOL)
                                                                                          of NEWVALUE]
        [ACCESSFNS STREAM (EXTERNALFORMAT.NAME (LISTGET (ffetch (STREAM OTHERPROPS) of DATUM)
                                                        'EXTERNALFORMAT.NAME)
                              (LET [(PROPS (ffetch (STREAM OTHERPROPS) of DATUM))
                                    (NAME (COND
                                             ((LITATOM NEWVALUE)
                                              NEWVALUE)
                                             (T (MKATOM NEWVALUE]
                                   (freplace (STREAM NOTXCCS) of DATUM with T)
                                   (COND
                                      (PROPS (LISTPUT PROPS 'EXTERNALFORMAT.NAME NAME))
                                      (T (freplace (STREAM OTHERPROPS) of DATUM with (LIST 'EXTERNALFORMAT.NAME
                                                                                          NAME]
        [ACCESSFNS STREAM (INCCODEFN (LET [(XFMT (LISTGET (ffetch (STREAM OTHERPROPS) of DATUM)
                                                          'EXTERNALFORMAT]
                                          (AND (type? EXTERNALFORMAT XFMT)
                                               (fetch (EXTERNALFORMAT INCCODEFN) of XFMT]
        [ACCESSFNS STREAM (PEEKCCODEFN (LET [(XFMT (LISTGET (ffetch (STREAM OTHERPROPS) of DATUM)
                                                            'EXTERNALFORMAT]
                                            (AND (type? EXTERNALFORMAT XFMT)
                                                 (fetch (EXTERNALFORMAT PEEKCCODEFN) of XFMT]
        [ACCESSFNS STREAM (BACKCHARFN (LET [(XFMT (LISTGET (ffetch (STREAM OTHERPROPS) of DATUM)
```

```
                                                                'EXTERNALFORMAT]
                                       (AND (type? EXTERNALFORMAT XFMT)
                                            (fetch (EXTERNALFORMAT BACKCHARFN) of XFMT]
         (ACCESSFNS STREAM (FILEOUTCHARFN (ffetch (STREAM OUTCHARFN) of DATUM)))
         (SYNONYM CBUFPTR (CPPTR))
         USERCLOSEABLE _ T USERVISIBLE _ T ACCESSBITS _ NoBits CPAGE _ 0 EPAGE _ 0 BUFFS _ NIL BYTESIZE _ 8
         CBUFPTR _ NIL MAXBUFFERS _ (LET NIL (DECLARE (GLOBALVARS \STREAM.DEFAULT.MAXBUFFERS))
                                             \STREAM.DEFAULT.MAXBUFFERS)
         CHARPOSITION _ 0 LINELENGTH _ (LET NIL (DECLARE (GLOBALVARS FILELINELENGTH))
                                             FILELINELENGTH)
         OUTCHARFN _ (FUNCTION \FILEOUTCHARFN)
         ENDOFSTREAMOP _ (FUNCTION \EOSERROR)
         IMAGEOPS _ \NOIMAGEOPS EOLCONVENTION _ (SELECTQ (SYSTEMTYPE)
                                                   (D CR.EOLC)
                                                   (VAX LF.EOLC)
                                                   (JERICHO CRLF.EOLC)
                                                   CR.EOLC)
         STRMBINFN _ (FUNCTION \STREAM.NOT.OPEN)
         STRMBOUTFN _ (FUNCTION \STREAM.NOT.OPEN)
         LASTCCODE _ 65535 NOTXCCS _ NIL)
)

(/DECLAREDATATYPE 'STREAM
      '(WORD WORD FLAG (BITS 3)
             POINTER BYTE BYTE WORD WORD WORD FLAG FLAG FLAG FLAG POINTER FLAG FLAG FLAG FLAG POINTER FLAG
             (BITS 2)
             FLAG POINTER POINTER POINTER WORD WORD POINTER POINTER POINTER POINTER POINTER WORD WORD WORD WORD
             POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER WORD WORD POINTER)

      ;; ---field descriptor list elided by lister---

      '56)

(DECLARE%: EVAL@COMPILE

(PUTPROPS STREAMOP MACRO [ARGS (CONS 'SPREADAPPLY* (CONS (COND
                                                          ((EQ (CAR (LISTP (CAR ARGS)))
                                                               'QUOTE)
                                                           (LIST 'fetch (CADAR ARGS)
                                                                 'of
                                                                 (CADR ARGS)))
                                                          (T (HELP "STREAMOP - OPNAME not quoted:" ARGS)))
                                                        (CDDR ARGS])
)

(DECLARE%: EVAL@COMPILE

(RPAQQ AppendBit 2)

(RPAQQ NoBits 0)

(RPAQQ ReadBit 1)

(RPAQQ WriteBit 4)

(RPAQ OutputBits (LOGOR AppendBit WriteBit))

(RPAQ BothBits (LOGOR ReadBit OutputBits))

(RPAQQ \NORUNCODE 255)

(CONSTANTS AppendBit NoBits ReadBit WriteBit (OutputBits (LOGOR AppendBit WriteBit))
      (BothBits (LOGOR ReadBit OutputBits))
      \NORUNCODE)
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS TestMasked MACRO ((BITS MASK)
                           (NEQ (LOGAND BITS MASK)
                                0)))

(PUTPROPS APPENDABLE MACRO ((STREAM)
                           (TestMasked (fetch ACCESSBITS of STREAM)
                                 AppendBit)))

(PUTPROPS APPENDONLY MACRO ((STREAM)
                           (EQ (fetch ACCESSBITS of STREAM)
                               AppendBit)))

(PUTPROPS DIRTYABLE MACRO [(STREAM)
                           (TestMasked (fetch ACCESSBITS of STREAM)
                                 (CONSTANT (LOGOR AppendBit WriteBit])

(PUTPROPS OPENED MACRO ((STREAM)
                        (NEQ (fetch ACCESSBITS of STREAM)
                             NoBits)))
```

```
(PUTPROPS OVERWRITEABLE MACRO ((STREAM)
                                 (TestMasked (fetch ACCESSBITS of STREAM)
                                           WriteBit)))

(PUTPROPS READABLE MACRO ((STREAM)
                            (TestMasked (fetch ACCESSBITS of STREAM)
                                 ReadBit)))

(PUTPROPS READONLY MACRO ((STREAM)
                            (EQ (fetch ACCESSBITS of STREAM)
                                ReadBit)))

(PUTPROPS WRITEABLE MACRO [(STREAM)
                             (OR (OVERWRITEABLE STREAM)
                                 (AND (APPENDABLE STREAM)
                                      (\EOFP STREAM])
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS \RUNCODED MACRO (OPENLAMBDA (STREAM)
                                       ;; returns NIL is the stream is not runcoded, that is, if the stream has 16 bit bytes explicitly represented
                                                        ; note that neq is ok since charsets are known to be SMALLP's
                                       (NEQ (fetch CHARSET of STREAM)
                                            \NORUNCODE)))
)

(RPAQQ EOLCONVENTIONS ((CR.EOLC 0)
                        (LF.EOLC 1)
                        (CRLF.EOLC 2)))

(DECLARE%: EVAL@COMPILE

(RPAQQ CR.EOLC 0)

(RPAQQ LF.EOLC 1)

(RPAQQ CRLF.EOLC 2)

(CONSTANTS (CR.EOLC 0)
       (LF.EOLC 1)
       (CRLF.EOLC 2))
)
)
```

;; END EXPORTED DEFINITIONS

```
(DEFINEQ
```

(STREAMPROP
```
  [LAMBDA X                                                    (* rda%: "22-Aug-84 14:24")
```
    ;; general top level entry for both fetching and setting stream properties.

```
    (COND
       ((IGREATERP X 2)
        (PUTSTREAMPROP (ARG X 1)
              (ARG X 2)
              (ARG X 3)))
       ((EQ X 2)
        (GETSTREAMPROP (ARG X 1)
              (ARG X 2)))
       (T (\ILLEGAL.ARG NIL])
```

(GETSTREAMPROP
```
  [LAMBDA (STREAM PROP)                                        (* rda%: "22-Aug-84 16:17")
    (OR (type? STREAM STREAM)
        (\ILLEGAL.ARG))
    (LISTGET (fetch (STREAM OTHERPROPS) of STREAM)
          PROP])
```

(PUTSTREAMPROP
```
  [LAMBDA (STREAM PROP VALUE)                                  (* rda%: "22-Aug-84 16:11")
    (OR (type? STREAM STREAM)
        (\ILLEGAL.ARG STREAM))
    (PROG ((OLDDATA (fetch OTHERPROPS of STREAM))
            OLDVALUE)
          (RETURN (PROG1 (COND
                            (OLDDATA (SETQ OLDVALUE (LISTGET OLDDATA PROP)))
                                  [COND
                                     (VALUE (LISTPUT OLDDATA PROP VALUE))
                                     (OLDVALUE                        ; Remove the property
                                              (COND
                                                 ((EQ (CAR OLDDATA)
                                                      PROP)
```

```
                                             (replace OTHERPROPS of STREAM with (CDDR OLDDATA)))
                                    (T (for TAIL on (CDR OLDDATA) by (CDDR TAIL)
                                              when (EQ (CADR TAIL)
                                                       PROP)
                                              do (FRPLACD TAIL (CDDDR TAIL))
                                                 (RETURN]
                         OLDVALUE)
                  (VALUE (replace OTHERPROPS of STREAM with (LIST PROP VALUE))
                                                                        ; know old value is NIL
                         NIL])
```

(**STREAMP**
```
  [LAMBDA (X)                                          (* rmk%: "14-OCT-83 14:35")
    (AND (type? STREAM X)
         X])
```
)

;; make streams print pretty

(DEFINEQ

(**DEFPRINT.BY.NAME**
```
  [LAMBDA (OBJECT STREAM NAME TYPENAME)                ; Edited  8-May-87 15:53 by bvm:

    ;; Print an object using its name, for example, #<FDev ERIS/76,5432>.  NAME is the object's name (or NIL if this one happens to be nameless),
    ;; TYPENAME is a string giving the generic name you want to appear in front, e.g., "FDev"

    [.SPACECHECK. STREAM (+ (NCHARS TYPENAME)
                            (PROGN                     ; Longest address is '< /177,177777>'
                                14)
                            (COND
                               (NAME (NCHARS NAME))
                               (T 0]
    (\OUTCHAR STREAM (fetch (READTABLEP HASHMACROCHAR) of *READTABLE*))
    (\OUTCHAR STREAM (CHARCODE <))
    (\SOUT (MKSTRING TYPENAME)
           STREAM)
    (COND
       (NAME (\OUTCHAR STREAM (CHARCODE SPACE))
             (\SOUT (MKSTRING NAME)
                    STREAM)))
    (\OUTCHAR STREAM (CHARCODE /))
    (\PRINTADDR OBJECT STREAM)
    (\OUTCHAR STREAM (CHARCODE >))
    T])
```

(**STREAM.DEFPRINT**
```
  [LAMBDA (STRM OUTSTREAM)                             ; Edited 19-Aug-88 14:01 by bvm
    (LET ((TYPE (SELECTC (fetch ACCESSBITS of STRM)
                    (ReadBit "Input")
                    (OutputBits "Output")
                    (BothBits "IO")
                    (AppendBit "Append")
                    "Closed")))
         (\DEFPRINT.BY.NAME STRM OUTSTREAM NIL (COND
                                                  ((fetch (STREAM NAMEDP) of STRM)
                                                                 ; Use file name
                                                   (CONCAT TYPE " Stream on " (fetch (STREAM FULLFILENAME)
                                                                                 of STRM)))
                                                  (T               ; Name the device
                                                   (CONCAT TYPE " " [CL:STRING-CAPITALIZE
                                                                      (STRING (fetch (FDEV DEVICENAME)
                                                                                 of (fetch DEVICE of STRM]
                                                           " Stream"])
```

(**FDEV.DEFPRINT**
```
  [LAMBDA (DEV STREAM)                                 ; Edited  8-May-87 15:55 by bvm

    ;; Print device using its name, for example, #<FDev ERIS/76,5432>

    (\DEFPRINT.BY.NAME DEV STREAM (fetch (FDEV DEVICENAME) of DEV)
           "FDev"])
```
)

(DECLARE%: DONTEVAL@LOAD DOCOPY

(DEFPRINT 'STREAM (FUNCTION \STREAM.DEFPRINT))

(DEFPRINT 'FDEV (FUNCTION \FDEV.DEFPRINT))
)

;; Needed because of STREAM initialization

(RPAQ? **FILELINELENGTH** 102)

(RPAQ? **\STREAM.DEFAULT.MAXBUFFERS** 3)

(DEFINEQ

GETACCESS
```
  [LAMBDA (STREAM)                                                    (* bvm%: "26-DEC-81 15:43")
```
    ;; Decodes the access bits. The inverse of the encoding in \SETACCESS. Ugly but no less so than the machinery to do it elegantly.

```
    (SELECTC (fetch ACCESSBITS of STREAM)
        (NoBits NIL)
        (ReadBit 'INPUT)
        (AppendBit 'APPEND)
        (OutputBits 'OUTPUT)
        (BothBits 'BOTH)
        (SHOULDNT))
```

SETACCESS
```
  [LAMBDA (STREAM ACCESS)                                             (* rmk%: " 7-NOV-83 15:02")
```
    ;; The setfn for the ACCESS field. Does not assume that streams are initialized with all bits off and \STREAM.NOT.OPEN installed

```
    (UNINTERRUPTABLY
        (PROG ((DEVICE (fetch DEVICE of STREAM)))
              (SELECTQ ACCESS
                  (NIL (replace ACCESSBITS of STREAM with NoBits)  ; Was open, now closing
                       (replace BINABLE of STREAM with (replace BOUTABLE of STREAM
                                                             with (replace EXTENDABLE of STREAM with NIL)))
                       (replace STRMBINFN of STREAM with (replace STRMBOUTFN of STREAM
                                                             with (FUNCTION \STREAM.NOT.OPEN))))
                  (INPUT (replace ACCESSBITS of STREAM with ReadBit)
                                                             ; Was closed, now opening
                       (replace BINABLE of STREAM with (fetch FDBINABLE of DEVICE))
                       (replace STRMBINFN of STREAM with (fetch BIN of DEVICE))
                       (replace STRMBOUTFN of STREAM with (FUNCTION \STREAM.NOT.OPEN))
                       (replace BOUTABLE of STREAM with (replace EXTENDABLE of STREAM with NIL)))
                  (APPEND (replace ACCESSBITS of STREAM with AppendBit)
                       (replace BOUTABLE of STREAM with (fetch FDBOUTABLE of DEVICE))
                       (replace EXTENDABLE of STREAM with (fetch FDEXTENDABLE of DEVICE))
                       (replace STRMBOUTFN of STREAM with (fetch BOUT of DEVICE))
                       (replace STRMBINFN of STREAM with (FUNCTION \STREAM.NOT.OPEN))
                       (replace BINABLE of STREAM with NIL))
                  (OUTPUT (replace ACCESSBITS of STREAM with OutputBits)
                       (replace BOUTABLE of STREAM with (fetch FDBOUTABLE of DEVICE))
                       (replace EXTENDABLE of STREAM with (fetch FDEXTENDABLE of DEVICE))
                       (replace STRMBOUTFN of STREAM with (fetch BOUT of DEVICE))
                       (replace STRMBINFN of STREAM with (FUNCTION \STREAM.NOT.OPEN))
                       (replace BINABLE of STREAM with NIL))
                  (BOTH (replace ACCESSBITS of STREAM with BothBits)
                       (replace BINABLE of STREAM with (fetch FDBINABLE of DEVICE))
                       (replace BOUTABLE of STREAM with (fetch FDBOUTABLE of DEVICE))
                       (replace EXTENDABLE of STREAM with (fetch FDEXTENDABLE of DEVICE))
                       (replace STRMBINFN of STREAM with (fetch BIN of DEVICE))
                       (replace STRMBOUTFN of STREAM with (fetch BOUT of DEVICE)))
                  (RAID "Illegal stream access mode"))))
        ACCESS])
)

(DECLARE%: DONTCOPY


;; FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

(PUTPROPS FDEVOP DMACRO [ARGS (LET ((OPNAME (CAR ARGS))
                                    (METHOD-DEVICE (CADR ARGS))
                                    (TAIL (CDDR ARGS)))
                                  (COND
                                     [(AND (LISTP OPNAME)
                                           (EQ (CAR OPNAME)
                                               'QUOTE))
                                      '(SPREADAPPLY* (fetch (FDEV ,(CADR OPNAME)) of ,METHOD-DEVICE)
                                             ,@TAIL]
                                     (T (ERROR "OPNAME not quoted: " OPNAME)])

(PUTPROPS \RECOGNIZE-HACK DMACRO [ARGS (LET ((NAME (CAR ARGS))
                                             (RECOG (CADR ARGS))
                                             (DEVICE (CADDR ARGS)))
                                           '(if (type? STREAM ,NAME)
                                                then ,NAME
                                              else (FDEVOP 'GETFILENAME ,DEVICE ,NAME ,RECOG ,DEVICE)])
)

(DECLARE%: EVAL@COMPILE
```

```
(DATATYPE FDEV ((RESETABLE FLAG)                                    ; Obsolete
                (RANDOMACCESSP FLAG)
                (NODIRECTORIES FLAG)
                (PAGEMAPPED FLAG)                                   ; True if i/o handled by pmap routines
                (FDBINABLE FLAG)                                    ; Copied as a microcode flag for INPUT streams formed on this
                                                                   ; device
                (FDBOUTABLE FLAG)
                (FDEXTENDABLE FLAG)
                (BUFFERED FLAG)                                     ; True implies that the device supports the BIN & BOUT uCode
                                                                   ; conventions, and implements the GETNEXTBUFFER method
                (DEVICENAME POINTER)                                ; Identifying name somehow
                (REMOTEP FLAG)                                      ; true if device not local to machine
                (SUBDIRECTORIES FLAG)                               ; true if device has real subdirectories
                (INPUT-INDIRECTED FLAG)                             ; True for devices that indirect their input stream.  Method
                                                                   ; INPUTSTREAM fetches it
                (OUTPUT-INDIRECTED FLAG)                            ; True for devices that indirect their output stream.  Method
                                                                   ; OUTPUTSTREAM fetches it
                (DEVICEINFO POINTER)                                ; arbitrary device-specific info stored here
                (OPENFILELST POINTER)                               ; Default place to keep list of streams open on this device
                ;; -----Rest of record consists of device "methods"-----
                ;; -----Following fields required of all devices-----
                (HOSTNAMEP POINTER)                                 ; (hostname {device}) => T if hostname is valid.  If device is
                                                                   ; given, return a FDEV for this {new} host, or T to use existing
                                                                   ; device
                (EVENTFN POINTER)                                   ; (device event), called before/after logout, sysout, makesys
                ;; -----Following fields required of all named devices, e.g., ones that open files-----
                (DIRECTORYNAMEP POINTER)                            ; (host/dir) => true if directory exists on host
                (OPENFILE POINTER)                                  ; (name access recog otherinfo device) => new stream open on
                                                                   ; this device, or NIL if name not found
                (CLOSEFILE POINTER)                                 ; (stream) => closes stream, returns it
                (REOPENFILE POINTER)                                ; (name access recog otherinfo device stream) like openfile, but
                                                                   ; called after logout to revalidate file, so optionally uses info in old
                                                                   ; stream to keep this opening like the previous
                (GETFILENAME POINTER)                               ; (name recog device) => full file name
                (DELETEFILE POINTER)                                ; (name) => deletes file so named, returning name, or NIL on
                                                                   ; failure.  RECOG=OLDEST
                (GENERATEFILES POINTER)                             ; (device pattern) => generator object for files matching pattern.
                                                                   ; Car of object is generator function, cdr is arbitrary state.
                                                                   ; Generator fn returns next file, or NIL when finished
                (RENAMEFILE POINTER)                                ; (olddevice oldfile newdevice newfile) to rename file on this
                                                                   ; (olddevice) to a potentially different device.
                (OPENP POINTER)                                     ; (name access dev) => stream if name is open for access, or all
                                                                   ; open streams if name = NIL
                (REGISTERFILE POINTER)                              ; (stream dev) => registers stream on its device
                (UNREGISTERFILE POINTER)                            ; (stream dev) => unregisters a stream from its device
                (FREEPAGECOUNT POINTER)                             ; (host/dir dev) => # of free pages on host/dir
                (MAKEDIRECTORY POINTER)                             ; (host/dir dev)
                (CHECKFILENAME POINTER)                             ; (name dev) => name if it is well-formed file name for dev
                (HOSTALIVEP POINTER)                                ; (host dev) => true if host is alive, i.e., responsive;  only defined
                                                                   ; if REMOTEP is true
                (BREAKCONNECTION POINTER)                           ; (host fastp dev) => closes connections to host
                ;; -----The following are required methods for operating on open streams-----
                (BIN POINTER)                                       ; (stream) => next byte of input
                (BOUT POINTER)                                      ; (stream byte) output byte to stream
                (PEEKBIN POINTER)                                   ; (stream) => next byte without advancing position in stream
                (READCHAR POINTER)                                  ; (stream) => next input char
                (WRITECHAR POINTER)                                 ; (stream char) => writes char to stream
                (PEEKCHAR POINTER)
                (UNREADCHAR POINTER)
                (READP POINTER)                                     ; (stream flag) => T if there is input available from stream right
                                                                   ; now
                (EOFP POINTER)                                      ; (stream) => T if BIN would signal eof.
                (BLOCKIN POINTER)                                   ; (stream buffer byteoffset nbytes)
                (BLOCKOUT POINTER)                                  ; (stream buffer byteoffset nbytes)
                (FORCEOUTPUT POINTER)                               ; (stream waitForFinish) flushes out to device anything that is
                                                                   ; buffered awaiting transmission
                (GETFILEINFO POINTER)                               ; (stream/name attribute device) => value of attribute for open
                                                                   ; stream or name of closed file
                (SETFILEINFO POINTER)                               ; (stream/name attribute newvalue device) sets attribute of open
                                                                   ; stream or closed file of given name
                (CHARSETFN POINTER)                                 ; (stream charset) => access function for the charset slot, for
                                                                   ; benefit of indirect streams.  See IMCHARSET for changing it on
                                                                   ; a file.
                (INPUTSTREAM POINTER)                               ; (stream) => indirected input stream
                (OUTPUTSTREAM POINTER)                              ; (stream) => indirected output stream
                ;; -----Following are required of random-access streams-----
                (GETFILEPTR POINTER)
                (GETEOFPTR POINTER)
                (SETFILEPTR POINTER)
                (BACKFILEPTR POINTER)                               ; (stream) backs up 'fileptr' by one.  Stream is only required to be
                                                                   ; able to do this once, i.e.  one-character buffer suffices
                (SETEOFPTR POINTER)                                 ; (stream length) => truncates or lengthens stream to indicated
                                                                   ; length
```

```
                (LASTC POINTER)                                       ; Should be possible only if RANDOMACCESSP
             ;; -----Following used for buffered streams-----
                (GETNEXTBUFFER POINTER)                               ; (stream whatfor noerrorflg) => Disposes of current buffer and
                                                                      ; optionally reads next.  whatfor is READ or WRITE.  Can cause
                                                                      ; EOF error unless noerrorflg
                (RELEASEBUFFER POINTER)                               ; (stream) => Does whatever appropriate when CBUFPTR is
                                                                      ; released
             ;; -----Following used for pagemapped streams-----
                (READPAGES POINTER)                                   ; (stream firstpage# buflist) => # of bytes read, starting at
                                                                      ; firstpage#, reading into buflist, a list of buffers or a single buffer
                                                                      ; (the usual case)
                (WRITEPAGES POINTER)                                  ; (stream firstpage# buflist) writes from buflist to stream starting
                                                                      ; at firstpage# of stream
                (TRUNCATEFILE POINTER)                                ; (stream page offset) make stream's eof be at page,offset,
                                                                      ; discarding anything after it
             ;; -----For window system, argh-----
                (WINDOWOPS POINTER)                                   ; window system operations
                (WINDOWDATA POINTER)                                  ; data for window systems
             ;; -----For any stream (here to not recompile everything)-----
                (READCHARCODE POINTER)                                ; Read a character code from the stream (cf BIN for bytes).

                )
        DIRECTORYNAMEP _ (FUNCTION NILL)
        HOSTNAMEP _ (FUNCTION NILL)
        READP _ (FUNCTION \GENERIC.READP)
        SETFILEPTR _ (FUNCTION \IS.NOT.RANDACCESSP)
        GETFILEPTR _ (FUNCTION \ILLEGAL.DEVICEOP)
        GETEOFPTR _ (FUNCTION \IS.NOT.RANDACCESSP)
        EOFP _ (FUNCTION \ILLEGAL.DEVICEOP)
        BLOCKIN _ (FUNCTION \GENERIC.BINS)
        BLOCKOUT _ (FUNCTION \GENERIC.BOUTS)
        RENAMEFILE _ (FUNCTION \GENERIC.RENAMEFILE)
        FORCEOUTPUT _ (FUNCTION NILL)
        REGISTERFILE _ (FUNCTION NILL)
        OPENP _ (FUNCTION NILL)
        UNREGISTERFILE _ (FUNCTION NILL)
        READCHAR _ (FUNCTION \GENERIC.READCHAR)
        WRITECHAR _ (FUNCTION \GENERIC.WRITECHAR)
        PEEKCHAR _ (FUNCTION \GENERIC.PEEKCHAR)
        UNREADCHAR _ (FUNCTION \GENERIC.UNREADCHAR)
        CHARSETFN _ (FUNCTION \GENERIC.CHARSET)
        BREAKCONNECTION _ (FUNCTION NILL)
        READCHARCODE _ (FUNCTION \GENERIC.READCCODE))

(RECORD FILEGENOBJ (NEXTFILEFN FILEINFOFN . GENFILESTATE))
)

(/DECLAREDATATYPE 'FDEV
        '(FLAG FLAG FLAG FLAG FLAG FLAG FLAG FLAG POINTER FLAG FLAG FLAG FLAG POINTER POINTER POINTER POINTER
                POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
                POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
                POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
                POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER)

        ;; ---field descriptor list elided by lister---

        '106)
)


;; END EXPORTED DEFINITIONS

(/DECLAREDATATYPE 'FDEV
        '(FLAG FLAG FLAG FLAG FLAG FLAG FLAG FLAG POINTER FLAG FLAG FLAG FLAG POINTER POINTER POINTER POINTER
                POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
                POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
                POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER
                POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER POINTER)

        ;; ---field descriptor list elided by lister---

        '106)

(ADDTOVAR SYSTEMRECLST
        (DATATYPE FDEV ((RESETABLE FLAG)
                        (RANDOMACCESSP FLAG)
                        (NODIRECTORIES FLAG)
                        (PAGEMAPPED FLAG)
                        (FDBINABLE FLAG)
                        (FDBOUTABLE FLAG)
                        (FDEXTENDABLE FLAG)
                        (BUFFERED FLAG)
                        (DEVICENAME POINTER)
                        (REMOTEP FLAG)
                        (SUBDIRECTORIES FLAG)
                        (INPUT-INDIRECTED FLAG)
```

```
                              (OUTPUT-INDIRECTED FLAG)
                              (DEVICEINFO POINTER)
                              (OPENFILELST POINTER)
                              (HOSTNAMEP POINTER)
                              (EVENTFN POINTER)
                              (DIRECTORYNAMEP POINTER)
                              (OPENFILE POINTER)
                              (CLOSEFILE POINTER)
                              (REOPENFILE POINTER)
                              (GETFILENAME POINTER)
                              (DELETEFILE POINTER)
                              (GENERATEFILES POINTER)
                              (RENAMEFILE POINTER)
                              (OPENP POINTER)
                              (REGISTERFILE POINTER)
                              (UNREGISTERFILE POINTER)
                              (FREEPAGECOUNT POINTER)
                              (MAKEDIRECTORY POINTER)
                              (CHECKFILENAME POINTER)
                              (HOSTALIVEP POINTER)
                              (BREAKCONNECTION POINTER)
                              (BIN POINTER)
                              (BOUT POINTER)
                              (PEEKBIN POINTER)
                              (READCHAR POINTER)
                              (WRITECHAR POINTER)
                              (PEEKCHAR POINTER)
                              (UNREADCHAR POINTER)
                              (READP POINTER)
                              (EOFP POINTER)
                              (BLOCKIN POINTER)
                              (BLOCKOUT POINTER)
                              (FORCEOUTPUT POINTER)
                              (GETFILEINFO POINTER)
                              (SETFILEINFO POINTER)
                              (CHARSETFN POINTER)
                              (INPUTSTREAM POINTER)
                              (OUTPUTSTREAM POINTER)
                              (GETFILEPTR POINTER)
                              (GETEOFPTR POINTER)
                              (SETFILEPTR POINTER)
                              (BACKFILEPTR POINTER)
                              (SETEOFPTR POINTER)
                              (LASTC POINTER)
                              (GETNEXTBUFFER POINTER)
                              (RELEASEBUFFER POINTER)
                              (READPAGES POINTER)
                              (WRITEPAGES POINTER)
                              (TRUNCATEFILE POINTER)
                              (WINDOWOPS POINTER)
                              (WINDOWDATA POINTER)
                              (READCHARCODE POINTER))))
```

;; EXTERNALFORMAT declaration and related functions

```
(DECLARE%: DOEVAL@COMPILE DONTCOPY
```

;; FOLLOWING DEFINITIONS EXPORTED

```
(DECLARE%: EVAL@COMPILE

(DATATYPE EXTERNALFORMAT ((EOLVALID FLAG)                      ; If true, the value of EOL field will replace the
                                                              ; EOLCONVENTION field of the resulted stream.
                          (EOL BITS 2)
                          (NIL BITS 1)
                          (INCCODEFN POINTER)                  ; Called with two arguments -- STREAM and COUNTP
                          (PEEKCCODEFN POINTER)                ; Called with three arguments -- STREAM, NOERROR and
                                                              ; COUNTP
                          (BACKCHARFN POINTER)                 ; Called with two arguments -- STREAM and COUNTP
                          (FILEOUTCHARFN POINTER)              ; Called with two arguments -- STREAM and CHARCODE

                          )
        EOLVALID _ NIL)
)

(/DECLAREDATATYPE 'EXTERNALFORMAT '(FLAG (BITS 2)
                                        (BITS 1)
                                        POINTER POINTER POINTER POINTER)
        ;; ---field descriptor list elided by lister---
        '8)
)
```

;; END EXPORTED DEFINITIONS

```
(/DECLAREDATATYPE 'EXTERNALFORMAT '(FLAG (BITS 2)
```

```
                                                  (BITS 1)
                                      POINTER POINTER POINTER POINTER)
        ;; ---field descriptor list elided by lister---

        '8)

(ADDTOVAR SYSTEMRECLST (DATATYPE EXTERNALFORMAT ((EOLVALID FLAG)
                                                 (EOL BITS 2)
                                                 (NIL BITS 1)
                                                 (INCCODEFN POINTER)
                                                 (PEEKCCODEFN POINTER)
                                                 (BACKCHARFN POINTER)
                                                 (FILEOUTCHARFN POINTER))))


(DEFINEQ
```

(\INSTALL.EXTERNALFORMAT
```
  [LAMBDA (NAME EXTERNALFORMAT)                                    ; Edited 26-Feb-91 17:33 by nm
```

;;; Install an external format vector, giving it the name NAME. EXTERNALFORMAT is an instance of the datatype EXTERNALFORMAT.
;;; *SUPPORTED-EXTERNALFORMATS* contains all installed external formats.   *EXTERNALFORMAT-NAME-TO-EXTERNALFORMAT* maps a
;;; name int its external format.

```
    (DECLARE (GLOBALVARS *SUPPORTED-EXTERNALFORMATS* *EXTERNALFORMAT-NAME-TO-EXTERNALFORMAT*))
    (COND
       ((type? EXTERNALFORMAT EXTERNALFORMAT)
        (LET (ENTRY)
            [SETQ NAME (COND
                          ((LITATOM NAME)
                            NAME)
                          (T  (MKATOM NAME]
            (UNINTERRUPTABLY
                [COND
                   ((SETQ ENTRY (FASSOC NAME *EXTERNALFORMAT-NAME-TO-EXTERNALFORMAT*))
                     (RPLACD ENTRY EXTERNALFORMAT))
                   (T  (pushnew *SUPPORTED-EXTERNALFORMATS* NAME)
                       (push *EXTERNALFORMAT-NAME-TO-EXTERNALFORMAT* (CONS NAME EXTERNALFORMAT]
                NAME)))
       (T (ERROR "INVALID EXTERNALFORMAT " EXTERNALFORMAT])
```

(\REMOVE.EXTERNALFORMAT
```
  [LAMBDA (EXTERNALFORMAT)                                         ; Edited 26-Feb-91 17:34 by nm
```

;;; Removes externalformat EXTERNALFORMAT and association between any of its name and EXTERNALFORMAT.

```
    (DECLARE (GLOBALVARS *SUPPORTED-EXTERNALFORMATS* *EXTERNALFORMAT-NAME-TO-EXTERNALFORMAT*))
    (LET (ENTRY)
        [COND
           ((SETQ ENTRY (CL:RASSOC EXTERNALFORMAT *EXTERNALFORMAT-NAME-TO-EXTERNALFORMAT*))
            (UNINTERRUPTABLY
               (SETQ *SUPPORTED-EXTERNALFORMATS* (DREMOVE (CAR ENTRY)
                                                          *SUPPORTED-EXTERNALFORMATS*))
               (SETQ *EXTERNALFORMAT-NAME-TO-EXTERNALFORMAT* (DREMOVE ENTRY
                                                          *EXTERNALFORMAT-NAME-TO-EXTERNALFORMAT*)))
           ]
        EXTERNALFORMAT])
```

(\GET.EXTERNALFORMAT.FROM.NAME
```
  [LAMBDA (NAME)
    (DECLARE (GLOBALVARS *EXTERNALFORMAT-NAME-TO-EXTERNALFORMAT*))
                                                                  ; Edited 26-Feb-91 17:33 by nm
    [SETQ NAME (COND
                  ((LITATOM NAME)
                    NAME)
                  (T  (MKATOM NAME]
    (CDR (FASSOC NAME *EXTERNALFORMAT-NAME-TO-EXTERNALFORMAT*])
```

(\EXTERNALFORMAT
```
  [LAMBDA (STREAM NEWVALUE)                                        ; Edited 26-Feb-91 13:20 by nm
```

;;; If NEWVALUE is nil, just returns the current external format name of STREAM. If NEWVALUE is supplied, the external format of STREAM is set to
;;; the external format named NEWVALUE.

```
    (\DTEST STREAM 'STREAM)
    (COND
       [NEWVALUE (COND
                    ((FMEMB NEWVALUE '(:XCCS :DEFAULT)
                         (freplace (STREAM NOTXCCS) of STREAM with NIL)))
                    [(FMEMB NEWVALUE *SUPPORTED-EXTERNALFORMATS*)
                     (freplace EXTERNALFORMAT.NAME of STREAM with NEWVALUE)
                     (freplace EXTERNALFORMAT of STREAM with (\DTEST (\GET.EXTERNALFORMAT.FROM.NAME NEWVALUE)
                                                          'EXTERNALFORMAT]
                    (T (ERROR NEWVALUE "INVALID EXTERNALFORMAT " NEWVALUE]
       (T (COND
```

```
                    ((ffetch (STREAM NOTXCCS) of STREAM)
                     (ffetch EXTERNALFORMAT.NAME of STREAM))
                    (T :DEFAULT])
)

(RPAQ? *SUPPORTED-EXTERNALFORMATS* '(:XCCS :DEFAULT))

(RPAQ? *EXTERNALFORMAT-NAME-TO-EXTERNALFORMAT* )

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS *SUPPORTED-EXTERNALFORMATS* *EXTERNALFORMAT-NAME-TO-EXTERNALFORMAT*)
)

(DEFINEQ
```

(**CREATE.JIS.EXTERNALFORMAT**
```
  [LAMBDA NIL                                                          ; Edited 25-Feb-91 17:21 by nm
```

;;; Create an instance of EXTERNALFORMAT datatype and install it with :JIS as its name.

```
    (LET [(XFMT (create EXTERNALFORMAT
                        INCCODEFN _ (FUNCTION \JISIN)
                        PEEKCCODEFN _ (FUNCTION \JISPEEK)
                        BACKCHARFN _ (FUNCTION \BACKJISCHAR)
                        FILEOUTCHARFN _ (FUNCTION \JISFILEOUTCHARFN]
        (\INSTALL.EXTERNALFORMAT :JIS XFMT])
```

(**CREATE.SHIFTJIS.EXTERNALFORMAT**
```
  [LAMBDA NIL                                                          ; Edited 25-Feb-91 18:15 by nm
```

;;; Create two instances of EXTERNALFORMAT datatype and install them with :W-MS and :MS as their names respectively. :MS have to change the
;;; end of line convention to CRLF.

```
    (LET ((XFMT1 (create EXTERNALFORMAT
                        INCCODEFN _ (FUNCTION \SHIFTJISIN)
                        PEEKCCODEFN _ (FUNCTION \SHIFTJISPEEK)
                        BACKCHARFN _ (FUNCTION \BACKSHIFTJISCHAR)
                        FILEOUTCHARFN _ (FUNCTION \SHIFTJISFILEOUTCHARFN)))
          (XFMT2 (create EXTERNALFORMAT
                        INCCODEFN _ (FUNCTION \SHIFTJISIN)
                        PEEKCCODEFN _ (FUNCTION \SHIFTJISPEEK)
                        BACKCHARFN _ (FUNCTION \BACKSHIFTJISCHAR)
                        FILEOUTCHARFN _ (FUNCTION \SHIFTJISFILEOUTCHARFN)
                        EOLVALID _ T
                        EOL _ CRLF.EOLC)))
        (\INSTALL.EXTERNALFORMAT :W-MS XFMT1)
        (\INSTALL.EXTERNALFORMAT :MS XFMT2])
```

(**CREATE.EUC.EXTERNALFORMAT**
```
  [LAMBDA NIL                                                          ; Edited 25-Feb-91 17:27 by nm
```

;;; Create an instance of EXTERNALFORMAT datatype and install it with :EUC as its name.

```
    (LET [(XFMT (create EXTERNALFORMAT
                        INCCODEFN _ (FUNCTION \EUCIN)
                        PEEKCCODEFN _ (FUNCTION \EUCPEEK)
                        BACKCHARFN _ (FUNCTION \BACKEUCCHAR)
                        FILEOUTCHARFN _ (FUNCTION \EUCFILEOUTCHARFN]
        (\INSTALL.EXTERNALFORMAT :EUC XFMT])
```

(**CREATE.THROUGH.EXTERNALFORMAT**
```
  [LAMBDA NIL                                                          ; Edited 26-Feb-91 13:33 by nm
```

;;; Create an instance of EXTERNALFORMAT datatype and install it with :THROUGH as its name. EOL is adjusted to CR so as not to do any eol
;;; conversion on this stream.

```
    (LET ((XFMT (create EXTERNALFORMAT
                        INCCODEFN _ (FUNCTION \THROUGHIN)
                        PEEKCCODEFN _ (FUNCTION \THROUGHPEEK)
                        BACKCHARFN _ (FUNCTION \BACKTHROUGHCHAR)
                        FILEOUTCHARFN _ (FUNCTION \THROUGHFILEOUTCHARFN)
                        EOLVALID _ T
                        EOL _ CR.EOLC)))
        (\INSTALL.EXTERNALFORMAT :THROUGH XFMT])
)

(DECLARE%: DONTEVAL@LOAD DOCOPY
```

(**CREATE.JIS.EXTERNALFORMAT**)

(**CREATE.SHIFTJIS.EXTERNALFORMAT**)

(**CREATE.EUC.EXTERNALFORMAT**)

(**CREATE.THROUGH.EXTERNALFORMAT**)
)

;; Device operations

(DEFINEQ

DEFINEDEVICE
```
[LAMBDA (NAME DEV)                                               (* bvm%: " 5-APR-83 15:33")
```
   ;; NIL DEV removes any device associated with NAME. NIL NAME simply adds the device without associating a name with it. This is useful for
   ;; getting its EVENTFN invoked. A litatom DEV makes NAME be a synonym for the device currently named DEV --- \FILEDEVICES contains each
   ;; device only once, \FILEDEVICENAMES contains each name device/host name only once (for spelling correction), and
   ;; \DEVICENAMETODEVICE maps a name into its device.

```
    (DECLARE (GLOBALVARS \FILEDEVICES \FILEDEVICENAMES \DEVICENAMETODEVICE))
    (PROG (TEMP)
          (SETQ NAME (U-CASE NAME))                              ; Use upper-case canonical device names
      RETRY
          (COND
            [(NULL DEV)
             (COND
                ((SETQ TEMP (FASSOC NAME \DEVICENAMETODEVICE))
                 (UNINTERRUPTABLY
                     (SETQ \DEVICENAMETODEVICE (DREMOVE TEMP \DEVICENAMETODEVICE))
                     (SETQ \FILEDEVICENAMES (DREMOVE NAME \FILEDEVICENAMES)))]
            [(type? FDEV DEV)
             (SETQ TEMP (FASSOC NAME \DEVICENAMETODEVICE))
             (UNINTERRUPTABLY
                 (COND
                    ((NOT (FMEMB DEV \FILEDEVICES))
                     [COND
                        (TEMP (SETQ \FILEDEVICES (DREMOVE (CDR TEMP)
                                                          \FILEDEVICES]
                                                          ; Devices are stored in inverse order of their definition, for proper
                                                          ; EVENTFN ordering.
                     (push \FILEDEVICES DEV)))
                 (COND
                    (NAME (pushnew \FILEDEVICENAMES NAME)
                          (RPLACD [OR TEMP (CAR (push \DEVICENAMETODEVICE (CONS NAME]
                                      DEV))))]
            ([AND (LITATOM DEV)
                  (SETQ TEMP (CDR (FASSOC (U-CASE DEV)
                                          \DEVICENAMETODEVICE]
             (SETQ DEV TEMP)
             (GO RETRY))
            (T (SETQ DEV (ERROR "INVALID FILE DEVICE" DEV))
               (GO RETRY)))
          (RETURN NAME])
```

GETDEVICEFROMNAME
```
  [LAMBDA (NAME NOERROR DONTCREATE)                              (* lmm " 5-Oct-84 18:06")
```
   ;; maps a filename (with host added) into a device

```
    (OR (AND (OR (LITATOM NAME)
                 (STRINGP NAME))
             (LET [(HOST (FILENAMEFIELD NAME 'HOST]
                  (\GETDEVICEFROMHOSTNAME (OR HOST NAME)
                       DONTCREATE)))
        (AND (NOT NOERROR)
             (LISPERROR "FILE NOT FOUND" NAME]
```

GETDEVICEFROMHOSTNAME
```
  [LAMBDA (HOSTN DONTCREATE)
    (DECLARE (GLOBALVARS \DEVICENAMETODEVICE \FILEDEVICES))      (* lmm " 5-Oct-84 14:36")
    (OR (CDR (FASSOC HOSTN \DEVICENAMETODEVICE))
        (CDR (FASSOC (SETQ HOSTN (U-CASE HOSTN))
                     \DEVICENAMETODEVICE))
        (AND (NOT DONTCREATE)
             (for D TEMP in \FILEDEVICES when (SETQ TEMP (FDEVOP 'HOSTNAMEP D HOSTN D))
                do ;; HOSTNAMEP is a pure predicate if the second arg is NIL.  Here we give a device, which indicates that we are not just a
                   ;; predicate, but in fact would like a new device back, possibly constructed from the old one.  A device value is installed with
                   ;; the new hostname;  a T value means install with D.

                   (COND
                      ((type? FDEV TEMP)
                       (SETQ D TEMP)))
                   (\DEFINEDEVICE HOSTN D)
                   (RETURN D])
```

REMOVEDEVICE
```
  [LAMBDA (DEV)                                                  (* bvm%: " 3-NOV-83 23:17")
```

```
;; Removes device DEV and also any association between any of its name and DEV
(DECLARE (GLOBALVARS \FILEDEVICES \FILEDEVICENAMES \DEVICENAMETODEVICE))
(PROG (TEMP)
      (UNINTERRUPTABLY
          (while (SETQ TEMP (find PAIR in \DEVICENAMETODEVICE suchthat (EQ (CDR PAIR)
                                                                          DEV)))
             do (SETQ \FILEDEVICENAMES (DREMOVE (CAR TEMP)
                                                \FILEDEVICENAMES))
                (SETQ \DEVICENAMETODEVICE (DREMOVE TEMP \DEVICENAMETODEVICE)))
          (SETQ \FILEDEVICES (DREMOVE DEV \FILEDEVICES)))
      (RETURN DEV])
```

(\REMOVEDEVICE.NAMES
```
  [LAMBDA (DEV NAMES)                                                     (* bvm%: "30-Jan-85 21:53")
     (DECLARE (GLOBALVARS \DEVICENAMETODEVICE \FILEDEVICENAMES))
```

;;; removes any names associated with device DEV without actually removing the device itself. If NAMES is non-NIL, removes only the names inside it

```
     (for TAIL on \DEVICENAMETODEVICE bind CHANGED when (AND (EQ (CDAR TAIL)
                                                                 DEV)
                                                            (OR (NULL NAMES)
                                                                (EQMEMB (CAAR TAIL)
                                                                        NAMES)))
        do (SETQ \FILEDEVICENAMES (DREMOVE (CAAR TAIL)
                                           \FILEDEVICENAMES))
           (RPLACA TAIL NIL)
           (SETQ CHANGED T)
        finally (COND
                  (CHANGED (SETQ \DEVICENAMETODEVICE (DREMOVE NIL \DEVICENAMETODEVICE])

)

(DEFINEQ
```

(\CLOSEFILE
```
  [LAMBDA (STREAM ABORTFLG)                                               ; Edited  8-May-87 16:35 by bvm
```

;; Close the file specified by the given open file descriptor and return the file handle.

```
     (COND
       ((NOT (READONLY STREAM))
        (IMAGEOP 'IMCLOSEFN STREAM STREAM)                               ; Do image-specific operations before physically closing the
                                                                          ; stream

        ))
     (LET ((DEVICE (fetch (STREAM DEVICE) of STREAM)))
          (PROG1 (FDEVOP 'CLOSEFILE DEVICE STREAM ABORTFLG)
                 (FDEVOP 'UNREGISTERFILE DEVICE DEVICE STREAM)
                 (replace (STREAM ACCESS) of STREAM with NIL)            ; This marks the STREAM as closed

                 )])
```

(\DELETEFILE
```
  [LAMBDA (FILENAME DEV)                                                  (* hdj "13-Jun-86 14:36")
     (SETQ FILENAME (\ADD.CONNECTED.DIR (\CONVERT-PATHNAME FILENAME)))
     (AND (OR DEV (SETQ DEV (\GETDEVICEFROMNAME FILENAME T)))
          (FDEVOP 'DELETEFILE DEV FILENAME DEV])
```

(\DEVICEEVENT
```
  [LAMBDA (EVENT)                                                         ; Edited 20-Aug-88 18:08 by bvm
```

;; Executes device-dependent event code so all devices can respond to various system transition events (LOGOUT, MAKESYS, etc.) Before an
;; event, devices are considered in the inverse order of their definition, so that older devices get processed later. The order is reversed for
;; after-events.

```
     (DECLARE (GLOBALVARS \FILEDEVICES))
     (LET ((BEFOREP (SELECTQ EVENT
                       ((BEFORELOGOUT BEFOREMAKESYS BEFORESYSOUT BEFORESAVEVM)
                        T)
                       NIL)))
          (for D in (if BEFOREP
                        then \FILEDEVICES
                      else (REVERSE \FILEDEVICES))
             do (FDEVOP 'EVENTFN D D EVENT)
                (if BEFOREP
                    then
```

;; Mark output files as needing revalidation if we write to them again. This is so that if you do a SAVEVM, then write to
;; the file some more, then boot back to the SAVEVM, that the AFTERSAVEVM event will notice that the stream has
;; changed.

;; Don't do this until AFTER we've run the eventfn because, e.g., the eventfn might have done a forceoutput on the
;; stream, thereby prematurely observing this flag.

```
                         (for STREAM in (FDEVOP 'OPENP D NIL 'OUTPUT D) unless (fetch (STREAM NONDEFAULTDATEFLG)
                                                                                      of STREAM)
                            do (replace (STREAM REVALIDATEFLG) of STREAM with T])
```

⟨\GENERATEFILES
```
[LAMBDA (PATTERN DESIREDPROPS OPTIONS)                        (* bvm%: "27-Apr-84 23:21")
```
;; Returns a file-generator object that will generate all files whose names match PATTERN. A gen-object consists of a device dependent
;; NEXTFILEFN and GENFILESTATE

```
(SETQ PATTERN (\ADD.CONNECTED.DIR PATTERN))
(PROG ((FDEV (\GETDEVICEFROMNAME PATTERN)))
      (RETURN (FDEVOP 'GENERATEFILES FDEV FDEV PATTERN DESIREDPROPS OPTIONS])
```

⟨\GENERATENEXTFILE
```
[LAMBDA (GENOBJ NAMEONLY)                                     (* bvm%: " 8-Jul-85 19:30")
```
;; GENOBJ is a file-generator object as created by \GENERATEFILES. The NEXTFILEFN must return the name of the next file generated by the
;; generator, as a string or symbol. Returns NIL if no files left. It updates GENOBJ so that it will get the following satisfactory file on the next call to
;; this function. --- If NAMEONLY, then filenames returned need not contain host, directory or version

```
(CL:FUNCALL (fetch NEXTFILEFN of GENOBJ)
            (fetch GENFILESTATE of GENOBJ)
            NAMEONLY])
```

⟨\GENERATEFILEINFO
```
[LAMBDA (GENOBJ ATTRIBUTE)                                    (* bvm%: "26-Apr-84 15:40")
```
;; GENOBJ is a file-generator object as created by \GENERATEFILES. The FILEINFOFN performs a GETFILEINFO on the file which is the
;; currently enumerated file, i.e., the last thing that NEXTFILEFN returned

```
(CL:FUNCALL (fetch FILEINFOFN of GENOBJ)
            (fetch GENFILESTATE of GENOBJ)
            ATTRIBUTE])
```

⟨\GETFILENAME
```
[LAMBDA (NAME RECOG FDEV)                                     (* hdj " 4-Sep-86 15:22")
```
;; Expands NAME according to recog, returning either the full NAME or NIL.

```
(SETQ NAME (\ADD.CONNECTED.DIR (\CONVERT-PATHNAME NAME)))
(COND
   ((OR FDEV (SETQ FDEV (\GETDEVICEFROMNAME NAME T)))
    (FDEVOP 'GETFILENAME FDEV NAME RECOG FDEV])
```

⟨\GENERIC.READCCODE
```
[LAMBDA (FILE RDTBL)                                          ; Edited 13-Jan-88 10:04 by jds
```
;;; returns a 16 bit character code. \INCHAR does the EOL conversion and this function converts to a 16 bit value. Saves the character for LASTC as
;;; well.

;; This is the GENERIC method for READCCODE.

```
(LET ((*READTABLE* (\GTREADTABLE RDTBL))
      (\RefillBufferFn (FUNCTION \READCREFILL))
      (STREAM (\INSTREAMARG FILE)))
     (DECLARE (SPECVARS *READTABLE* \RefillBufferFn))
     (replace (STREAM LASTCCODE) of STREAM with (\INCCODE STREAM]))
```

⟨\GENERIC.OUTFILEP
```
[LAMBDA (NAME DEV)                                            (* lmm " 6-Jan-85 17:41")
 (PROG ((V (FDEVOP 'GETFILENAME DEV NAME 'OLD DEV)))
       (RETURN (if V
                   then (PACKFILENAME 'VERSION (ADD1 (OR (FILENAMEFIELD V 'VERSION)
                                                          1))
                                      'BODY V)
                   else (PACKFILENAME 'VERSION 1 'BODY NAME])
```

⟨\OPENFILE
```
[LAMBDA (NAME ACCESS RECOG PARAMETERS)                        (* hdj "14-Oct-86 14:04")
```
;;; Opens the file identified by NAME possibly expanded according to RECOG. Returns an open stream for the file. ACCESS is assumed to be one of
;;; INPUT, OUTPUT, BOTH, or APPEND.

```
(PROG (FDEV CDNAME STREAM)
  RETRY
      [COND
         [(type? STREAM NAME)
          (COND
             ((\IOMODEP NAME ACCESS T)
              (\DO.PARAMS.AT.OPEN NAME ACCESS PARAMETERS)
              (RETURN NAME))
             (T (SETQ CDNAME NAME)
                (SETQ FDEV (fetch (STREAM DEVICE) of NAME]
         (T (SETQ CDNAME (\ADD.CONNECTED.DIR NAME))
            (SETQ FDEV (\GETDEVICEFROMNAME CDNAME]          ; Keep NAME for possible error
```
;; The OPENFILE operation returns NIL if the file wasn't found, so the name is right for the not-found error. That error must not be generated from
;; inside the device, or spellfile would be too constrained. The won't-open error may happen inside the device, if the device itself does some
;; interlocking (e.g. a file-server). The generic code in OPENFILE may also generate that error, to enforce interlocks among files already opened

```
        ;; in this Lisp.
                (COND
                    ((SETQ STREAM (FDEVOP 'OPENFILE FDEV CDNAME ACCESS RECOG PARAMETERS FDEV))
                     (replace ACCESS of STREAM with ACCESS)
                     (replace CPAGE of STREAM with (COND
                                                        ((EQ ACCESS 'APPEND)
                                                         (fetch EPAGE of STREAM))
                                                        (T 0)))
                     (replace COFFSET of STREAM with (COND
                                                        ((EQ ACCESS 'APPEND)
                                                         (fetch EOFFSET of STREAM))
                                                        (T 0)))
                     (\DO.PARAMS.AT.OPEN STREAM ACCESS PARAMETERS)

                     ;; register the file using its internal device's registerfile method instead of FDEV's; this is primarily for the benefit of the file cacher

                     (LET ((STREAM-FDEV (fetch (STREAM DEVICE) of STREAM)))
                          (FDEVOP 'REGISTERFILE STREAM-FDEV STREAM-FDEV STREAM))
                     (RETURN STREAM))
                    (T (SETQ NAME (LISPERROR "FILE NOT FOUND" NAME))
                       (GO RETRY)])
```

\DO.PARAMS.AT.OPEN
```
  [LAMBDA (STREAM ACCESS PARAMETERS)                                          ; Edited  5-Oct-92 13:45 by jds

    ;; Does generic parameters when a file/stream is open.  Called by \OPENFILE and OPENSTREAM

    (for X ATTR VAL in PARAMETERS do (COND
                                        [(LISTP X)
                                         (SETQ ATTR (CAR X))
                                         (SETQ VAL (CAR (LISTP (CDR X]
                                        (T (SETQ ATTR X)
                                           (SETQ VAL T)))
                                     (SELECTQ ATTR
                                        (BUFFERS (SETFILEINFO STREAM 'BUFFERS VAL))
                                        (ENDOFSTREAMOP
                                            (SETFILEINFO STREAM 'ENDOFSTREAMOP VAL))
                                        (CHARSET (CHARSET STREAM VAL))
                                        (EXTERNALFORMAT
                                            (\EXTERNALFORMAT STREAM VAL))
                                        (CONVHANKAKU (CONVHANKAKU STREAM VAL))
                                        ((EOL EOLCONVENTION EOLC)
                                            (replace EOLCONVENTION of STREAM with (SELECTQ VAL
                                                                                    (CR CR.EOLC)
                                                                                    (LF LF.EOLC)
                                                                                    (CRLF CRLF.EOLC)
                                                                                    (\ILLEGAL.ARG VAL))))
                                        NIL])
```

\RENAMEFILE
```
  [LAMBDA (OLDFILE NEWFILE)                                                    (* hdj " 7-May-86 12:22")
    (SETQ OLDFILE (\ADD.CONNECTED.DIR OLDFILE))
    (SETQ NEWFILE (\ADD.CONNECTED.DIR NEWFILE))
    (LET ((OLD-DEVICE (\GETDEVICEFROMNAME OLDFILE T))
          (NEW-DEVICE (\GETDEVICEFROMNAME NEWFILE T)))
        (AND OLD-DEVICE (FDEVOP 'RENAMEFILE OLD-DEVICE OLD-DEVICE OLDFILE NEW-DEVICE NEWFILE])
```

\REVALIDATEFILE
```
  [LAMBDA (STREAM)                                                            (* bvm%: "30-DEC-81 17:45")

    ;; Check the file to determine if it corresponds to the status information for it found in the STREAM and file handle.  Return DELETED if the file no
    ;; longer exists, CHANGED if the file does not correspond to the status information, or NIL if everything is OK.

    (PROG ((NEWSTREAM (FDEVOP 'REOPENFILE (fetch DEVICE of STREAM)
                             (fetch FULLFILENAME of STREAM)
                             (fetch ACCESS of STREAM)
                             'OLD NIL (fetch DEVICE of STREAM)
                             STREAM)))
          (RETURN (COND
                    ((NOT NEWSTREAM)
                     'DELETED)
                    ((EQ NEWSTREAM STREAM)                                    ; Nothing changed
                     NIL)
                    (T (replace F1 of STREAM with (fetch F1 of NEWSTREAM))
                                                                             ; Copy 'device' information from the new opening to the old
                       (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))
                       (COND
                          ((EQUAL (fetch VALIDATION of NEWSTREAM)
                                  (fetch VALIDATION of STREAM))
                           NIL)
                          (T (replace VALIDATION of STREAM with (fetch VALIDATION of NEWSTREAM))
                             (replace EPAGE of STREAM with (fetch EPAGE of NEWSTREAM))
```

```
                                     (replace EOFFSET of STREAM with (fetch EOFFSET of NEWSTREAM))
                                     'CHANGED])
```

(**PAGED.REVALIDATEFILELST**
```
  [LAMBDA (DEVICE)                                                     (* hdj "30-Sep-86 15:23")
```

;;; Revalidate all of the open files on DEVICE (a PMAP device)

```
      (bind REASON PAGES for STREAM in (FDEVOP 'OPENP DEVICE NIL NIL DEVICE)
         do (if (SETQ REASON (\PAGED.REVALIDATEFILE STREAM))
                then (SELECTQ REASON
                          (CHANGED                                              ; it changed

                                                                               ; update the map
                                 (SETQ PAGES (RESTOREMAP STREAM)))
                          (DELETED                                             ; the file disappeared, so zap the stream
                                 (SETQ PAGES (FORGETPAGES STREAM))
                                 [MAPC (STREAMPROP STREAM 'AFTERCLOSE)
                                       (FUNCTION (LAMBDA (FN)
                                                     (APPLY* FN STREAM]
                                 (replace ACCESS of STREAM with NIL)
                                 (FDEVOP 'UNREGISTERFILE DEVICE DEVICE STREAM))
                          (SHOULDNT))
                       (\PRINT-REVALIDATION-RESULT REASON STREAM)))   ; might as well return something useful
      (FDEVOP 'OPENP DEVICE NIL NIL DEVICE])
```

(**PAGED.REVALIDATEFILES**
```
  [LAMBDA (LIST)                                                       (* hdj "30-Sep-86 15:18")
```

;;; Revalidate all of the open files on LIST; they are all PMAPped streams

```
      (LET ((NEWLIST (COPY LIST)))
           (bind REASON PAGES for STREAM in LIST
              do (if (SETQ REASON (\PAGED.REVALIDATEFILE STREAM))
                     then (SELECTQ REASON
                               (CHANGED                                       ; it changed - update the map
                                      (SETQ PAGES (RESTOREMAP STREAM)))
                               (DELETED                                       ; the file disappeared, so zap the stream
                                      (SETQ PAGES (FORGETPAGES STREAM))
                                      [MAPC (STREAMPROP STREAM 'AFTERCLOSE)
                                            (FUNCTION (LAMBDA (FN)
                                                          (APPLY* FN STREAM]
                                      (replace ACCESS of STREAM with NIL)
                                      (LET ((DEVICE (fetch (STREAM DEVICE) of STREAM)))
                                           (FDEVOP 'UNREGISTERFILE DEVICE DEVICE STREAM))
                                      (SETQ NEWLIST (DREMOVE STREAM NEWLIST)))
                               (SHOULDNT))
                            (\PRINT-REVALIDATION-RESULT REASON STREAM)))
```

;;; return the remaining files

```
           NEWLIST])
```

(**PAGED.REVALIDATEFILE**
```
  [LAMBDA (STREAM)                                                     (* hdj "23-May-86 14:14")
```
   ;; Check the file to determine if it corresponds to the status information for it found in the STREAM and file handle. Return DELETED if the file no
   ;; longer exists, CHANGED if the file does not correspond to the status information, or NIL if everything is OK
```
      (LET ((NEWSTREAM (FDEVOP 'REOPENFILE (fetch DEVICE of STREAM)
                               (fetch FULLFILENAME of STREAM)
                               (fetch ACCESS of STREAM)
                               'OLD NIL (fetch DEVICE of STREAM)
                               STREAM)))
           (COND
              ((NOT NEWSTREAM)
               'DELETED)
              ((EQ NEWSTREAM STREAM)                                     ; Nothing changed
               NIL)
              (T (replace F1 of STREAM with (fetch F1 of NEWSTREAM))    ; Copy 'device' information from the new opening to the old
                 (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))
                 (COND
                    ((EQUAL (fetch VALIDATION of NEWSTREAM)
                            (fetch VALIDATION of STREAM))
                     NIL)
                    (T (replace VALIDATION of STREAM with (fetch VALIDATION of NEWSTREAM))
                       (replace EPAGE of STREAM with (fetch EPAGE of NEWSTREAM))
                       (replace EOFFSET of STREAM with (fetch EOFFSET of NEWSTREAM))
                       'CHANGED])
```

◊\BUFFERED.REVALIDATEFILE
```
  [LAMBDA (STREAM)                                                          (* hdj "23-May-86 14:14")
    ;; Check the file to determine if it corresponds to the status information for it found in the STREAM and file handle.  Return DELETED if the file no
    ;; longer exists, CHANGED if the file does not correspond to the status information, or NIL if everything is OK

    (LET ((NEWSTREAM (FDEVOP 'REOPENFILE (fetch DEVICE of STREAM)
                                         (fetch FULLFILENAME of STREAM)
                                         (fetch ACCESS of STREAM)
                                         'OLD NIL (fetch DEVICE of STREAM)
                                         STREAM)))
        (COND
           ((NOT NEWSTREAM)
            'DELETED)
           ((EQ NEWSTREAM STREAM)                                           ; Nothing changed
            NIL)
           (T (replace F1 of STREAM with (fetch F1 of NEWSTREAM))          ; Copy 'device' information from the new opening to the old
              (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))
              (COND
                 ((EQUAL (fetch VALIDATION of NEWSTREAM)
                         (fetch VALIDATION of STREAM))
                  NIL)
                 (T (replace VALIDATION of STREAM with (fetch VALIDATION of NEWSTREAM))
                    (replace EPAGE of STREAM with (fetch EPAGE of NEWSTREAM))
                    (replace EOFFSET of STREAM with (fetch EOFFSET of NEWSTREAM))
                    'CHANGED)])
```

◊\BUFFERED.REVALIDATEFILELST
```
  [LAMBDA (DEVICE)                                                          (* hdj "30-Sep-86 15:16")

;;; Revalidate all of the open files on DEVICE (a buffered device)

    [bind REASON for STREAM in (FDEVOP 'OPENP DEVICE NIL NIL DEVICE)
       do (if (SETQ REASON (\BUFFERED.REVALIDATEFILE STREAM))
              then (SELECTQ REASON
                       ((DELETED CHANGED)                                   ; the file changed or disappeared, so zap the stream
                        [MAPC (STREAMPROP STREAM 'AFTERCLOSE)
                              (FUNCTION (LAMBDA (FN)
                                          (APPLY* FN STREAM]
                        (replace ACCESS of STREAM with NIL)
                        (FDEVOP 'UNREGISTERFILE DEVICE DEVICE STREAM)
                        (\PRINT-REVALIDATION-RESULT REASON STREAM))
                       (SHOULDNT]
    ;; might as well return something useful

    (FDEVOP 'OPENP DEVICE NIL NIL DEVICE])
```

◊\PRINT-REVALIDATION-RESULT
```
  [LAMBDA (RESULT STREAM)                                                   (* hdj "26-May-86 15:46")
    (printout T T T "**** WARNING:  The file " (fetch (STREAM FULLNAME) of STREAM))
    (SELECTQ RESULT
        (CHANGED (printout T " has been modified since you last accessed it!" T))
        (DELETED (printout T " was previously opened but has disappeared!" T))
        (SHOULDNT])
```

◊\TRUNCATEFILE
```
  [LAMBDA (STREAM LASTPAGE LASTOFFSET)                                      (* bvm%: " 8-MAY-82 16:11")
    ;; Shorten an open file to have the given last page and offset.  Last page = NIL means to truncate to the current length, which some devices may
    ;; interpret as a noop

    (FDEVOP 'TRUNCATEFILE (fetch DEVICE of STREAM)
            STREAM LASTPAGE LASTOFFSET])
```

◊\FILE-CONFLICT
```
  [LAMBDA (NAME ACCESS DEVICE)                                              ; Edited 14-Apr-87 18:07 by jop
    ;; returns NIL if there's no conflict between the access mode of the file we're about to open and the ones already open there's no conflict if there
    ;; are none already open, or if the ones already open are open for input, and so's the candidate

    (LET* ((FILENAME (if (type? STREAM NAME)
                         then (fetch (STREAM FULLFILENAME) of NAME)
                         else NAME))
           (STREAMS-FOR-THIS-FILE (FDEVOP 'OPENP DEVICE FILENAME NIL DEVICE)))
        (if STREAMS-FOR-THIS-FILE
            then [LET [(EXISTING-ACCESS-MODE (fetch (STREAM ACCESS) of (CAR STREAMS-FOR-THIS-FILE]
                     (if (NEQ ACCESS EXISTING-ACCESS-MODE)
                         then T
                       elseif (EQ ACCESS 'INPUT)
                         then NIL
```

```
                            else (NEQ NAME (CAR STREAMS-FOR-THIS-FILE]
            else NIL])
)
```

;; Generic enumerator

```
(DEFINEQ
```

(\GENERATENOFILES

```
  [LAMBDA (FDEV PATTERN DESIREDPROPS OPTIONS)                  (* bvm%: " 5-Jun-84 16:31")

    ;; A dummy function to be used by devices that don't support directory generation.  This produces a generate that generates no files.

    (PROG ((STAR (STRPOS '* PATTERN))
           (ESC (STRPOS '(CONSTANT (CHARACTER (CHARCODE ESC)))
                        PATTERN)))
         (RETURN (COND
                    ([AND [OR (NULL STAR)
                              (AND (EQ (NTHCHARCODE PATTERN (SUB1 STAR))
                                       (CHARCODE ;))
                                   (NULL (STRPOS '* PATTERN (ADD1 STAR]
                         (OR (NULL ESC)
                             (AND (EQ (NTHCHARCODE PATTERN (SUB1 ESC))
                                      (CHARCODE ;))
                                  (NULL (STRPOS (CONSTANT (CHARACTER (CHARCODE ESC)))
                                                PATTERN
                                                (ADD1 ESC]
                      (create FILEGENOBJ
                              NEXTFILEFN _ (FUNCTION \NOFILESNEXTFILEFN)
                              FILEINFOFN _ (FUNCTION \NOFILESINFOFN)
                              GENFILESTATE _ (create NOFILEGENSTATE
                                                     NOFILETYPE _ (COND
                                                                    ((AND (NULL STAR)
                                                                          (NULL ESC))
                                                                     'NOSTAR)
                                                                    (T (SETQ PATTERN (PACKFILENAME 'VERSION NIL
                                                                                                   'BODY PATTERN))
                                                                       'STAR))
                                                     NOFILEPATTERN _ PATTERN)))
                    (T (\NULLFILEGENERATOR])
```

(\NULLFILEGENERATOR

```
  [LAMBDA NIL                                                  (* bvm%: " 5-Jun-84 15:46")

    ;; A file generator that generates no files

    (create FILEGENOBJ
            NEXTFILEFN _ (FUNCTION NILL])
```

(\NOFILESNEXTFILEFN

```
  [LAMBDA (GENFILESTATE NAMEONLY)                              (* bvm%: " 8-Jul-85 19:28")
    (PROG (FILE TYPE)
          [SELECTQ (SETQ TYPE (fetch NOFILETYPE of GENFILESTATE))
              (NOSTAR (replace NOFILETYPE of GENFILESTATE with 'DONE)
                      (SETQ FILE (INFILEP (fetch NOFILEPATTERN of GENFILESTATE))))
              (DONE (RETURN NIL))
              (STAR
                  ;; Star in version field.  Start out by producing the oldest file, and note its version and the version of the newest file for
                  ;; subsequent enumeration

                  (SETQ FILE (FULLNAME (fetch NOFILEPATTERN of GENFILESTATE)
                                       'OLDEST))
                  [replace NOFILETYPE of GENFILESTATE with (CONS (FILENAMEFIELD FILE 'VERSION)
                                                                 (FILENAMEFIELD (INFILEP (fetch NOFILEPATTERN
                                                                                                of GENFILESTATE))
                                                                                'VERSION])
                  (PROG [(VER (ADD1 (CAR TYPE]

                     ;; TYPE is a dotted pair of versions (old . newest) -- test INFILEP for each version number after old until we get to newest

                    LP  (COND
                           ((IGREATERP VER (CDR TYPE))
                            (RETURN NIL))
                           [[SETQ FILE (INFILEP (PACKFILENAME.STRING 'VERSION VER 'BODY (fetch NOFILEPATTERN
                                                                                              of GENFILESTATE]
                            (RPLACA TYPE (FILENAMEFIELD FILE 'VERSION]
                           (T (add VER 1)
                              (GO LP]
          (RETURN (COND
                     (FILE (replace NOFILENAME of GENFILESTATE with FILE)
                           FILE])
```

(\NOFILESINFOFN

```
  [LAMBDA (GENSTATE ATTRIBUTE)                                 (* bvm%: "27-Apr-84 22:17")
```

;;; Fileinfo fn for getting attributes of the file currently enumerated -- go thru the generic GETFILEINFO

```
        (GETFILEINFO (fetch NOFILENAME of GENSTATE)
                 ATTRIBUTE])
)

(DECLARE%: DONTCOPY

(DECLARE%: EVAL@COMPILE

(RECORD NOFILEGENSTATE (NOFILETYPE NOFILEPATTERN . NOFILENAME))
)
)

(DEFINEQ
```

(**FILE.NOT.OPEN**
```
  [LAMBDA (X NOERROR)                                          (* hdj "17-Jun-86 18:28")

    ;; Returns NIL of NOERROR, otherwise causes the FILE NOT OPEN error.  Used by \GETSTREAM.  \STREAM.NOT.OPEN doesn't take
    ;; NOERROR arg.

    (AND (NULL NOERROR)
         (LISPERROR "FILE NOT OPEN" (COND
                                       ((type? STREAM X)
                                        (fetch (STREAM FULLNAME) of X))
                                       (T X)])
```

(**FILE.WONT.OPEN**
```
  [LAMBDA (X)                                                  (* hdj "17-Jun-86 18:32")
    (LISPERROR "FILE WON'T OPEN" (COND
                                   ((type? STREAM X)
                                    (fetch (STREAM FULLNAME) of X))
                                   (T X)])
```

(**ILLEGAL.DEVICEOP**
```
  [LAMBDA N                                                    (* bvm%: "28-DEC-81 15:44")
    (ERROR "Attempt to use undefined device operation" (for I from 1 to N collect (ARG N I])
```

(**IS.NOT.RANDACCESSP**
```
  [LAMBDA N                                                    (* hdj "17-Jun-86 18:32")
    (PROG ((THING (ARG N 1)))
          (RETURN (ERROR "File is not RANDACCESSP" (COND
                                                     ((type? STREAM THING)
                                                      (fetch (STREAM FULLNAME) of THING))
                                                     (T THING])
```

(**STREAM.NOT.OPEN**
```
  [LAMBDA (STREAM)                                             (* hdj "17-Jun-86 18:32")

    ;; Can be used as BIN/BOUT function.  \FILE.NOT.OPEN accepts more than just a stream, and also has NOERROR control

    (LISPERROR "FILE NOT OPEN" (fetch (STREAM FULLNAME) of STREAM])
)

(ADDTOVAR \FILEDEVICES )

(ADDTOVAR \FILEDEVICENAMES )

(ADDTOVAR \DEVICENAMETODEVICE )
```

;; Device instances

```
(DEFINEQ
```

(**FDEVINSTANCE**
```
  [LAMBDA (FDEV)                                               (* gbn "16-Sep-85 18:09")

    ;; Creates an 'instance' of FDEV, a distinct device that executes all the operations of FDEV, but which can be smashed to change those operations
    ;; in order to specialize streams. --- \INHERITFDEVOP.S and .D are macros that expect the device to be found from a STREAM or FDEV
    ;; argument, respectively.  Only operations that relate to streams are included, since non-stream device operations will be obtained from the original
    ;; device, whose name is registered.

    (create FDEV using FDEV DEVICEINFO _ FDEV CLOSEFILE _ (\INHERITFDEVOP.S CLOSEFILE STREAM)
                       GETFILEINFO _ (\INHERITFDEVOP.D GETFILEINFO STREAM ATTRIB FDEV)
                       OPENFILE _ (\INHERITFDEVOP.D OPENFILE CDNAME ACCESS RECOG OTHERINFO FDEV)
                       READPAGES _ (\INHERITFDEVOP.S READPAGES STREAM FIRSTPAGE BUFFERLIST)
                       SETFILEINFO _ (\INHERITFDEVOP.D SETFILEINFO STREAM ATTRIBUTE VALUE FDEV)
                       TRUNCATEFILE _ (\INHERITFDEVOP.S TRUNCATEFILE STREAM LASTPAGE LASTOFFSET)
                       WRITEPAGES _ (\INHERITFDEVOP.S WRITEPAGES STREAM FIRSTPAGE BUFFERLIST)
                       REOPENFILE _ (\INHERITFDEVOP.D REOPENFILE NAME ACCESS RECOG OTHERINFO FDEV OLDSTREAM)
                       BIN _ (\INHERITFDEVOP.S BIN STREAM)
                       BOUT _ (\INHERITFDEVOP.S BOUT STREAM BYTE)
                       PEEKBIN _ (\INHERITFDEVOP.S PEEKBIN STREAM NOERRORFLG)
                       BACKFILEPTR _ (\INHERITFDEVOP.S BACKFILEPTR STREAM)
                       SETFILEPTR _ (\INHERITFDEVOP.S SETFILEPTR STREAM INDX)
```

```
                              GETFILEPTR _ (\INHERITFDEVOP.S GETFILEPTR STREAM)
                              GETEOFPTR _ (\INHERITFDEVOP.S GETEOFPTR STREAM)
                              EOFP _ (\INHERITFDEVOP.S EOFP STREAM)
                              BLOCKIN _ (\INHERITFDEVOP.S BLOCKIN STREAM BASE OFFSET NBYTES)
                              BLOCKOUT _ (\INHERITFDEVOP.S BLOCKOUT STREAM BASE OFFSET NBYTES)
                              FORCEOUTPUT _ (\INHERITFDEVOP.S FORCEOUTPUT STREAM])
)

(DECLARE%: EVAL@COMPILE

(PUTPROPS \INHERITFDEVOP.D MACRO [X (SUBPAIR '(NEWARGS OPNAME . ARGS)
                                            (CONS (SUBST '(fetch DEVICEINFO of FDEV)
                                                         'FDEV
                                                         (CDR X))
                                                  X)
                                            '(FUNCTION (LAMBDA ARGS
                                                     (FDEVOP 'OPNAME (fetch DEVICEINFO of FDEV) . NEWARGS])

(PUTPROPS \INHERITFDEVOP.S MACRO [(OPNAME . ARGS)
                                     (FUNCTION (LAMBDA ARGS
                                              (FDEVOP 'OPNAME (fetch DEVICEINFO of (fetch DEVICE of STREAM))
                                                                                     . ARGS]
)
)

(RPAQ? LOGINHOST/DIR '{DSK})

(RPAQ? \CONNECTED.DIRECTORY '{DSK})

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS LOGINHOST/DIR \CONNECTED.DIRECTORY \FILEDEVICES \FILEDEVICENAMES \DEVICENAMETODEVICE)
)
```

;; Directory defaulting

```
(DEFINEQ
```

(**CNDIR**
```
  [LAMBDA (HOST/DIR)                                                    ; Edited 11-Mar-87 14:28 by Pavel
```

;;; Connects to HOST/DIR, verifying that HOST/DIR exists.

```
    (DECLARE (GLOBALVARS \CONNECTED.DIRECTORY))
    (LET ([TEMP-DEFAULTS (PATHNAME (SETQ \CONNECTED.DIRECTORY (OR (DIRECTORYNAME (AND HOST/DIR (
                                                                                        \CONVERT-PATHNAME
                                                                                         HOST/DIR))
                                                                                T
                                                                                'ASK)
                                                                   (ERROR "Non-existent directory" HOST/DIR]
          (NEW-DEFAULTS (COPY-PATHNAME *DEFAULT-PATHNAME-DEFAULTS*)))
        (CL:SETF (%%PATHNAME-HOST NEW-DEFAULTS)
               (CL:PATHNAME-HOST TEMP-DEFAULTS))
        (CL:SETF (%%PATHNAME-DEVICE NEW-DEFAULTS)
               (CL:PATHNAME-DEVICE TEMP-DEFAULTS))
        (CL:SETF (%%PATHNAME-DIRECTORY NEW-DEFAULTS)
               (CL:PATHNAME-DIRECTORY TEMP-DEFAULTS))
        (SETQ *DEFAULT-PATHNAME-DEFAULTS* NEW-DEFAULTS))
    \CONNECTED.DIRECTORY])
```

(**DIRECTORYNAME**
```
  [LAMBDA (DIRNAME STRPTR CREATE?)                                      ; Edited 3-Sep-91 15:18 by jrb:
```

    ;; Returns connected directory name

```
    (AND (CL:PATHNAMEP DIRNAME)
         (SETQ DIRNAME (CL:NAMESTRING DIRNAME)))
    (SELECTQ (SYSTEMTYPE)
        (VAX (GETDIRNAME))
        (D (DECLARE (GLOBALVARS LOGINHOST/DIR))
           [PROG (DN FDEV)
                 [SELECTQ DIRNAME
                     (T                                       ; Connected host/dir
                        (SETQ DN \CONNECTED.DIRECTORY))
                     (NIL (SETQ DN (OR LOGINHOST/DIR '{DSK})))
                     (COND
                        [(AND [SETQ FDEV (LET [(HOST (FILENAMEFIELD DIRNAME 'HOST]
                                              (\GETDEVICEFROMHOSTNAME
                                                 (OR HOST (FILENAMEFIELD (SELCHARQ (NTHCHARCODE DIRNAME 1)
                                                                           ((< /)
                                                                            (SETQ DIRNAME
                                                                              (PACKFILENAME.STRING
                                                                               'DIRECTORY DIRNAME
                                                                               'BODY \CONNECTED.DIRECTORY))
)
                                                                           (SETQ DIRNAME (PACKFILENAME.STRING
```

```
                                                                              'SUBDIRECTORY
                                                                              DIRNAME 'DIRECTORY
                                                                              \CONNECTED.DIRECTORY
                                                                              )))
                                                          'HOST]
                            (SETQ DN (FDEVOP 'DIRECTORYNAMEP FDEV DIRNAME FDEV CREATE?)))
                      (COND
                          ((EQ DN T)
                           (SETQ DN (PACKFILENAME.STRING 'HOST (fetch (FDEV DEVICENAME) of FDEV)
                                          'DIRECTORY DIRNAME]
                      (T (RETURN]
                   (RETURN (if (NOT STRPTR)
                               then (MKSTRING DN)
                             elseif (EQ STRPTR T)
                               then (MKATOM DN)
                               else (MKSTRING DN])
              (HELP])
```

(**DIRECTORYNAMEP**
```
  [LAMBDA (DIRNAME HOSTNAME)                                     (* bvm%: "18-Oct-85 14:38")
    ;; T if DIRNAME is recognized as a currently existing directory, on HOSTNAME, or if not included, on the hostname in DIRNAME, or the connected
    ;; host.

    (LET ([DN (COND
                (HOSTNAME (PACKFILENAME.STRING 'DIRECTORY DIRNAME 'HOST HOSTNAME))
                (T (PACKFILENAME.STRING 'DIRECTORY DIRNAME 'DIRECTORY \CONNECTED.DIRECTORY]
          FDEV)
         (AND (SETQ FDEV (\GETDEVICEFROMNAME DN T))
              (FDEVOP 'DIRECTORYNAMEP FDEV DN FDEV)
              T])
```

(**HOSTNAMEP**
```
  [LAMBDA (NAME)                                                  (* rmk%: "11-NOV-81 14:33")
    ;; T if NAME is the name of a recognizable host

    (DECLARE (GLOBALVARS \FILEDEVICENAMES \FILEDEVICES))
    (PROG (N)
         (COND
            ((LITATOM NAME)
             (SETQ N (U-CASE NAME)))
            [(STRINGP NAME)
             (SETQ N (MKATOM (U-CASE NAME]
            (T (RETURN NIL)))
         [COND
            ((EQ (CHCON1 N)
                 (CHARCODE {))
             (SETQ N (SUBATOM N 2 (SUB1 (OR (STRPOS '} N 2)
                                            (RETURN NIL]
         (RETURN (AND (OR (MEMB N \FILEDEVICENAMES)
                          (find D in \FILEDEVICES suchthat (FDEVOP 'HOSTNAMEP D N)))
                      T])
```

(**ADD.CONNECTED.DIR**
```
  [LAMBDA (FILENAME)                                              ; Edited 29-Dec-89 15:41 by jds
    ;; Modifies the filename to include connected host and/or dir

    (COND
       ([AND (OR (LITATOM FILENAME)
                 (STRINGP FILENAME))
             (NOT (UNPACKFILENAME.STRING FILENAME 'HOST]
        (PACKFILENAME.STRING 'BODY FILENAME 'DIRECTORY \CONNECTED.DIRECTORY))
       (T FILENAME))
)
```

```
;; Binary I/O Public functions

(DEFINEQ
```

(**BACKFILEPTR**
```
  [LAMBDA (STREAM)                                                (* bvm%: "30-JAN-82 16:59")
    (FDEVOP 'BACKFILEPTR (fetch DEVICE of STREAM)
            STREAM])
```

(**BACKPEEKBIN**
```
  [LAMBDA (STREAM)                                                (* bvm%: " 7-Jun-84 16:45")
    ;; Returns previous byte on file without changing fileptr.  Returns NIL if we are positioned at the beginning of the file.  Called by LASTC

    (UNINTERRUPTABLY
        (AND (\BACKFILEPTR STREAM)
             (\BIN STREAM)))])
```

(\BACKBIN
```
[LAMBDA (STREAM)                                                   (* bvm%: " 7-Jun-84 16:46")
```
;; Returns previous character on file and backs up fileptr so that next \BIN will also return it. Returns NIL if we are positioned at the beginning of the
;; file.

```
(AND (\BACKFILEPTR STREAM)
     (\PEEKBIN STREAM])
```

(\BIN
```
[LAMBDA (STREAM)                                                   (* lmm "20-APR-82 22:00")
                                                                  ; MERELY EXECUTE OPCODE

    (\BIN STREAM])
```

(\BIN
```
[LAMBDA (STREAM)                                                   (* rmk%: " 2-NOV-83 14:32")
                                                                  ; UFN for BIN opcode

    (STREAMOP 'STRMBINFN (SETQ STREAM (\DTEST STREAM 'STREAM))
          STREAM])
```

(\BINS
```
[LAMBDA (STREAM BASE OFF NBYTES)                                   (* bvm%: "25-MAY-83 12:48")
```
;; BINs NBYTES bytes from STREAM into BASE+OFF

```
    (FDEVOP 'BLOCKIN [ffetch DEVICE of (SETQ STREAM (\DTEST STREAM 'STREAM]
          STREAM BASE OFF NBYTES])
```

(BOUT
```
[LAMBDA (STREAM BYTE)                                              ; Edited  3-Mar-87 16:04 by lal
                                                                  ; Merely execute opcode

    (if (NUMBERP BYTE)
        then (if (GREATERP BYTE 65535)
                 then (\ILLEGAL.ARG BYTE)))
    (\BOUT STREAM BYTE])
```

(\BOUT
```
[LAMBDA (STREAM BYTE)                                              ; Edited  8-Jan-88 17:00 by jds
    [COND
        ((NUMBERP BYTE)
         (COND
            ((GREATERP BYTE 65535)
             (\ILLEGAL.ARG BYTE]
    (SETQ STREAM (\DTEST STREAM 'STREAM))
    (STREAMOP 'STRMBOUTFN STREAM STREAM BYTE])
```

(\BOUTS
```
[LAMBDA (STREAM BASE OFF NBYTES)                                   (* bvm%: "25-MAY-83 12:47")
```
;; BOUTs NBYTES bytes from BASE+OFF into OFD. Follows logic of BINS.

```
    (FDEVOP 'BLOCKOUT [ffetch DEVICE of (SETQ STREAM (\DTEST STREAM 'STREAM]
          STREAM BASE OFF NBYTES])
```

(COPYBYTES
```
[LAMBDA (SRCFIL DSTFIL START END)                                 ; Edited 24-Jun-88 15:08 by drc:
```
;; Copies bytes from START up to but not including END from SRCFIL into DSTFIL.

```
    (PROG ((SRC (\GETSTREAM SRCFIL 'INPUT))
           (DST (\GETSTREAM DSTFIL 'OUTPUT))
           NBYTES)
          (SETQ NBYTES (COND
                         (END                                     ; Specified a start and ending
                              (COND
                                ((EQUAL START END)                ; special case: no bytes to copy
                                 (RETURN)))
                              [\SETFILEPTR SRC (COND
                                                ((type? BYTEPTR START)
                                                 START)
                                                (T (\ILLEGAL.ARG START]
                              (IDIFFERENCE (COND
                                             [(EQ END -1)
                                              (COND
                                                ((RANDACCESSP SRC)
                                                                  ; It's random access, so GETEOFPTR will work
                                                 (\GETEOFPTR SRC))
                                                (T                ; Otherwise, we have to hack around this (probably a bug in FTP
                                                                  ; streams)
                                                 (GETFILEINFO SRC 'LENGTH]
                                             ((type? BYTEPTR END)
                                              END)
                                             (T (\ILLEGAL.ARG END)))
                                           START))
```

```
                            (T START)))                            ; How much to copy, or NIL if to EOF
                    (COND
                        ((AND NBYTES (ILESSP NBYTES 0))
                         (ERROR "Negative number of bytes to copy" NBYTES)))
                    [COND
                        ((fetch BUFFERED of (fetch DEVICE of SRC))    ; Can copy by the bufferfull
                         (\BUFFERED.COPYBYTES SRC DST NBYTES))
                        [[OR NBYTES (SETQ NBYTES (COND
                                                    ((fetch RANDOMACCESSP of (fetch DEVICE of SRC))
                                                     (IDIFFERENCE (\GETEOFPTR SRC)
                                                                  (\GETFILEPTR SRC]
                                                    ; Know how many bytes to copy
                         (FRPTQ NBYTES (\BOUT DST (\BIN SRC]
                        (T                                            ; Copying to EOF but can't tell when that will happen
                            (until (\EOFP SRC) do (\BOUT DST (\BIN SRC]
                    (RETURN T)                                        ; As specified in VM

                ])
```

(**COPYCHARS**
```
  [LAMBDA (SRCFIL DSTFIL START END)                              (* rmk%: "11-Mar-85 12:12")

    ;; This is similar to COPYBYTES except that conversion is done between the EOL convention of the input and the EOL convention of the output

    (PROG ((SRCSTRM (\GETSTREAM SRCFIL))
           (DSTSTRM (\GETSTREAM DSTFIL))
           (ACTUALSTART 0)
          RAP ACTUALEND EOF SRCEOLC DSTEOLC CH)
         [COND
             ((EQ (SETQ SRCEOLC (fetch EOLCONVENTION of SRCSTRM))
                  (SETQ DSTEOLC (fetch EOLCONVENTION of DSTSTRM)))
              (RETURN (COPYBYTES SRCSTRM DSTSTRM START END]
         [COND
             ((SETQ RAP (fetch RANDOMACCESSP of (fetch DEVICE of SRCSTRM)))
              (SETQ EOF (\GETEOFPTR SRCSTRM]
         (COND
             [END (OR RAP (ERROR "COPYCHARS: Source file is not random access" (fetch FULLFILENAME of SRCSTRM)))
                  (OR (type? BYTEPTR (SETQ ACTUALSTART (FIX START)))
                      (LISPERROR "ILLEGAL ARG" START))
                  (\SETFILEPTR SRCSTRM ACTUALSTART)
                  (SETQ ACTUALEND (COND
                                      ((EQ END -1)
                                       EOF)
                                      ((type? BYTEPTR END)
                                       END)
                                      (T (\ILLEGAL.ARG END]
             [START (SETQ ACTUALEND (COND
                                        (RAP (SETQ ACTUALSTART (\GETFILEPTR SRCSTRM))
                                             (IMIN EOF (IPLUS START ACTUALSTART)))
                                        (T START]
             (RAP (SETQ ACTUALSTART (\GETFILEPTR SRCSTRM))
                  (SETQ ACTUALEND EOF))
             (T (until (\EOFP SRCSTRM) do (\OUTCHAR DSTSTRM (\INCHAR SRCSTRM)))
                                                    ; Not RAP and START and END are both NIL.  Slow copy to the
                                                    ; end of the file.
                (RETURN)))
         (OR (IGEQ ACTUALEND ACTUALSTART)
             (ERROR "Negative number of bytes to copy" (IDIFFERENCE ACTUALEND ACTUALSTART)))
                                                    ; We only have to worry about mismatched EOLCs
         (SELECTC SRCEOLC
             (CR.EOLC                                             ; DST is either CRLF or LF
                    (FRPTQ (IDIFFERENCE ACTUALEND ACTUALSTART)
                           (SELCHARQ (SETQ CH (\BIN SRCSTRM))
                               (CR (AND (EQ DSTEOLC CRLF.EOLC)
                                        (\BOUT DSTSTRM (CHARCODE CR)))
                                   (\BOUT DSTSTRM (CHARCODE LF)))
                               (\BOUT DSTSTRM CH))))
             (LF.EOLC                                             ; DST is either CRLF or CR
                    (FRPTQ (IDIFFERENCE ACTUALEND ACTUALSTART)
                           (SELCHARQ (SETQ CH (\BIN SRCSTRM))
                               (LF (\BOUT DSTSTRM (CHARCODE CR))
                                   (AND (EQ DSTEOLC CRLF.EOLC)
                                        (\BOUT DSTSTRM (CHARCODE LF))))
                               (\BOUT DSTSTRM CH))))
             (CRLF.EOLC                                           ; DST is either CR or LF
                   [for I from (IDIFFERENCE ACTUALEND ACTUALSTART) to 1 by -1
                      do (\BOUT DSTSTRM (COND
                                           ((OR (NEQ (SETQ CH (\BIN SRCSTRM))
                                                     (CHARCODE CR))
                                                (EQ I 1))
                                            CH)
                                           [(PROGN (add I -1)
                                                    ; Adjust for second character
                                                   (EQ (SETQ CH (\BIN SRCSTRM))
                                                       (CHARCODE LF)))
                                            (COND
                                                ((EQ DSTEOLC CR.EOLC)
```

```
                                                                (CHARCODE CR))
                                                           (T (CHARCODE LF]
                                                  (T (\BOUT DSTSTRM (CHARCODE CR))
                                                     CH])
                        (SHOULDNT)))
        T])
```

(**COPYFILE**
```
  [LAMBDA (FROMFILE TOFILE DESTPARAMETERS)                            ; Edited  2-Jan-93 13:35 by jds
```

;;; DESTPARAMETERS is like PARAMETERS arg to OPENSTREAM -- overrides default parameters

```
    [AND (DIRECTORYNAMEP (PACKFILENAME 'HOST NIL 'BODY TOFILE)
              (UNPACKFILENAME TOFILE 'HOST))
          (SETQ TOFILE (PACKFILENAME 'DIRECTORY TOFILE 'BODY (PACKFILENAME 'HOST NIL 'DIRECTORY NIL 'BODY
                                                                                 FROMFILE]
    (RESETLST
        [RESETSAVE [SETQ FROMFILE (OPENSTREAM FROMFILE 'INPUT 'OLD '((SEQUENTIAL T)
                                                                       (DON'TCACHE T]
                '(PROGN (CLOSEF OLDVALUE]
        (\COPYOPENFILE FROMFILE TOFILE DESTPARAMETERS))])
```

(**COPYOPENFILE**
```
  [LAMBDA (INSTREAM NEWNAME DESTPARAMETERS)                           ; Edited 17-Sep-90 11:41 by jds
                                                                     (* bvm%: "18-Oct-85 15:54")
  (PROG ((PROPS DESTPARAMETERS)
          TYPE X OUTSTREAM)
        [COND
            ([AND (NOT (ASSOC 'CREATIONDATE DESTPARAMETERS))
                  (SETQ X (GETFILEINFO INSTREAM 'CREATIONDATE]
              (push PROPS (LIST 'CREATIONDATE X]
        [COND
            ((SETQ X (GETFILEINFO INSTREAM 'LENGTH))
              (push PROPS (LIST 'LENGTH X]
        [COND
            [(SETQ TYPE (CADR (ASSOC 'TYPE DESTPARAMETERS]
            ((OR (AND (SETQ TYPE (GETFILEINFO INSTREAM 'TYPE))
                      (NEQ TYPE '?))
                 (SETQ TYPE (\INFER.FILE.TYPE INSTREAM)))
              (push PROPS (LIST 'TYPE TYPE]
```
   ;; TAL removed : (COND ((AND (EQ TYPE 'TEXT) (SETQ X (GETFILEINFO INSTREAM 'EOL)) (NOT (ASSOC 'EOL DESTPARAMETERS)))
   ;; (push PROPS (LIST 'EOL X)))) --- if the caller didn't specify, we ought to convert to the destination system's EOL convention for text files.
```
        [RESETSAVE [SETQ OUTSTREAM (OPENSTREAM NEWNAME 'OUTPUT 'NEW `((SEQUENTIAL T)
                                                                      (DON'TCACHE T)
                                                                      ,@PROPS]
                '(AND RESETSTATE (SETQ OLDVALUE (CLOSEF OLDVALUE))
                      (DELFILE OLDVALUE]
```
   ;; Obsoleted by Lyric's multiple streams: (OR (EQ (\GETFILEPTR INSTREAM) 0) (SETFILEPTR INSTREAM 0)) ;; In case it was open by
   ;; someone else! Really need multiple streams, but until then at least don't lose big this way
```
        (COND
            [(AND (EQ TYPE 'TEXT)
                  (NEQ (GETFILEINFO OUTSTREAM 'EOL)
                       X))                                          ; Incompatible EOL conventions, do slow way
              (replace ENDOFSTREAMOP of INSTREAM with (FUNCTION NILL))
              (bind CH (SRCEOL _ (fetch EOLCONVENTION of INSTREAM)) until (NULL (SETQ CH (\BIN INSTREAM)))
                 do (\OUTCHAR OUTSTREAM (\CHECKEOLC CH SRCEOL INSTREAM]
            (T (COPYBYTES INSTREAM OUTSTREAM)))
```
   ;; On UNIX version, give FX the option of printing a warning when the file type is defaulted -- they want to discourage that behavior.
```
        (AND (EQ \MACHINETYPE \MAIKO)
             FileTypeConfirmFlg
             (STKPOS 'COPYFILE)
             (NULL (ASSOC 'TYPE DESTPARAMETERS))
             (\UFStoOtherCopyMess INSTREAM OUTSTREAM))
```
   ;; We return the closed stream.
```
        (RETURN (CLOSEF OUTSTREAM])
```

(**INFER.FILE.TYPE**
```
  [LAMBDA (STREAM)                                                   (* bvm%: " 8-Jun-84 11:48")
```
   ;; STREAM is open on a file whose TYPE is unknown. If we can, decide between TEXT and BINARY by examining bytes
```
  (COND
      ((RANDACCESSP STREAM)
       (SETFILEPTR STREAM 0)
       (PROG ((OLDEOF (fetch ENDOFSTREAMOP of STREAM))
              TYPE)
             (replace ENDOFSTREAMOP of STREAM with (FUNCTION NILL))
             [SETQ TYPE (do (COND
                                ((IGREATERP (OR (\BIN STREAM)
                                                (RETURN 'TEXT))
                                   127)
```

```
                                                (RETURN 'BINARY]
                        (replace ENDOFSTREAMOP of STREAM with OLDEOF)
                        (SETFILEPTR STREAM 0)                                    ; Put file ptr back
                        (RETURN TYPE])
```

(EOFP
```
  [LAMBDA (FILE)                                                        (* bvm%: "10-Jun-84 22:46")

    ;; User entry.  T if FILE is at EOF.  I-10 only considers input files, we merely give priority to them

    (\EOFP (OR (\GETSTREAM FILE 'INPUT T)
               (\GETSTREAM FILE])
```

(FORCEOUTPUT
```
  [LAMBDA (STREAM WAITFORFINISH)                                        (* bvm%: "27-Apr-84 22:45")
    (SETQ STREAM (\GETSTREAM STREAM 'OUTPUT))
    (FDEVOP 'FORCEOUTPUT (fetch DEVICE of STREAM)
            STREAM WAITFORFINISH])
```

(\FLUSH.OPEN.STREAMS
```
  [LAMBDA (FDEV)                                                        (* hdj " 5-Jun-86 12:58")
```

;;; flush all of device's open streams

```
    (for STREAM in (\DEVICE-OPEN-STREAMS FDEV) when (DIRTYABLE STREAM) do (FDEVOP 'FORCEOUTPUT
                                                                                  (fetch (STREAM DEVICE)
                                                                                     of STREAM)
                                                                                  STREAM])
```

(CHARSET
```
  [LAMBDA (STREAM NEWVALUE)                                             ; Edited 11-Sep-87 16:22 by bvm:

    ;; Public access to a stream's CHARSET.  If NEWVALUE is given, changes the charset (which for output streams can write a charset shift).  We
    ;; invoke the stream's device's get/set charset method on the stream, and also invoke the IMCHARSET image operation (which is where file
    ;; streams get to write a charset shift).

    ;; If CHARACTERSET is either 255 or T, set the stream so that it's non run-coded, i.e., you read 2 bytes for each character read.

    (SETQ STREAM (\GETSTREAM STREAM))
    (COND
        ((EQ NEWVALUE NSCHARSETSHIFT)                                   ; Coerce 255 to T for uniformity
         (SETQ NEWVALUE T))
        ([NOT (OR (EQ NEWVALUE NIL)
                  (EQ NEWVALUE T)
                  (AND (>= NEWVALUE 0)
                       (< NEWVALUE \MAXCHARSET]
         (\ILLEGAL.ARG NEWVALUE)))
    (LET [(OLDVAL (ACCESS-CHARSET STREAM (if (EQ NEWVALUE T)
                                             then NSCHARSETSHIFT
                                             else NEWVALUE]              ; First modify the stream's slot
        (if (EQ OLDVAL NSCHARSETSHIFT)
            then (SETQ OLDVAL T))
        (if (AND NEWVALUE (NEQ OLDVAL NEWVALUE))
            then                                                         ; Now invoke the imageop if anything interesting happened
                (IMAGEOP 'IMCHARSET STREAM STREAM NEWVALUE))
        OLDVAL])
```

(ACCESS-CHARSET
```
  [LAMBDA (STREAM NEWVALUE)                                             ; Edited 11-Sep-87 15:46 by bvm:
    (FDEVOP 'CHARSETFN (fetch (STREAM DEVICE) of STREAM)
            STREAM NEWVALUE])
```

(GETEOFPTR
```
  [LAMBDA (FILE)                                                        (* rmk%: "21-OCT-83 11:19")
    (PROG ((STREAM (\GETSTREAM FILE)))
          (RETURN (FDEVOP 'GETEOFPTR (fetch DEVICE of STREAM)
                          STREAM])
```

(GETFILEINFO
```
  [LAMBDA (FILE ATTRIB)                                                 ; Edited  8-May-87 16:53 by bvm
    (LET (FULLNAME DEV)
         (COND
             [(type? STREAM FILE)                                       ; FILE is open or nameless.  Ask device for info;  if it can't handle
                                                                        ; it, at least handle some generic cases
              (COND
                  ((EQ ATTRIB 'ACCESS)
                   (fetch ACCESS of FILE))
                  ((FDEVOP 'GETFILEINFO (SETQ DEV (fetch DEVICE of FILE))
                           FILE ATTRIB DEV))
                  ((OPENED FILE)                                        ; Could be false for a closed nameless stream
                   (SELECTQ ATTRIB
                       ((BYTESIZE OPENBYTESIZE)
                        (fetch BYTESIZE of FILE))
```

```
                         (EOL (SELECTC (fetch EOLCONVENTION of FILE)
                                 (CR.EOLC 'CR)
                                 (LF.EOLC 'LF)
                                 (CRLF.EOLC 'CRLF)
                                 (SHOULDNT)))
                         (BUFFERS (fetch MAXBUFFERS of FILE))
                         (CHARSET (CHARSET FILE))
                         (ENDOFSTREAMOP
                             (fetch ENDOFSTREAMOP of FILE))
                         (LENGTH (AND (RANDACCESSP FILE)
                                 (\GETEOFPTR FILE)))
                         (SIZE [SIZE.FROM.LENGTH (OR (FDEVOP 'GETFILEINFO DEV FILE 'LENGTH DEV)
                                                 (AND (RANDACCESSP FILE)
                                                     (\GETEOFPTR FILE])
                    NIL))
               ((EQ ATTRIB 'SIZE)
                (SIZE.FROM.LENGTH (FDEVOP 'GETFILEINFO DEV FILE 'LENGTH DEV]
           [(AND [SETQ DEV (\GETDEVICEFROMNAME (SETQ FULLNAME (\ADD.CONNECTED.DIR (\CONVERT-PATHNAME FILE]
                 (SETQ FULLNAME (FDEVOP 'GETFILENAME DEV FULLNAME 'OLD DEV)))
                                                            ; Name of existing file.  It's possible we should have the device
                                                            ; do recognition instead, but then we have the confusion of file
                                                            ; not found recovery in the wrong place.
             (SELECTQ ATTRIB
                 ((ACCESS OPENBYTESIZE)                      ; Strip off attributes that apply only to open files
                     NIL)
                 (OR (FDEVOP 'GETFILEINFO DEV FULLNAME ATTRIB DEV)
                     (SELECTQ ATTRIB
                         (SIZE (SIZE.FROM.LENGTH (FDEVOP 'GETFILEINFO DEV FULLNAME 'LENGTH DEV)))
                         NIL]
           (T (LISPERROR "FILE NOT FOUND" FILE))
```

(\TYPE.FROM.FILETYPE
```
  [LAMBDA (FILETYPE)                                       (* bvm%: "15-Jan-85 16:22")
```

;;; Coerces a numeric FILETYPE to a symbolic TYPE or returns FILETYPE itself if it is not registered on the list FILING.TYPES

```
    (AND FILETYPE (OR (CAR (find PAIR in FILING.TYPES suchthat (EQ (CADR PAIR)
                                                                 FILETYPE)))
                 FILETYPE])
```

(\FILETYPE.FROM.TYPE
```
  [LAMBDA (TYPE)                                           (* bvm%: "15-Jan-85 17:08")
    (OR (CADR (ASSOC TYPE FILING.TYPES))
        (FIXP TYPE])
```

(GETFILEPTR
```
  [LAMBDA (FILE)                                           (* rmk%: "21-OCT-83 11:19")
    (PROG ((STREAM (\GETSTREAM FILE)))
          (RETURN (FDEVOP 'GETFILEPTR (fetch DEVICE of STREAM)
                      STREAM])
```

(SETFILEINFO
```
  [LAMBDA (FILE ATTRIB VALUE)                              ; Edited 27-Mar-89 15:33 by bvm
    (LET (FULLNAME DEV)
         (COND
             [(type? STREAM FILE)                          ; FILE is open, so strip off attributes that can be set from the
                                                           ; stream.
              (SELECTQ ATTRIB
                  ((ACCESS BYTESIZE OPENBYTESIZE)          ; These can't be changed for an open file
                      NIL)
                  (EOL (replace EOLCONVENTION of FILE with (SELECTQ VALUE
                                                             (CR CR.EOLC)
                                                             (CRLF CRLF.EOLC)
                                                             (LF LF.EOLC)
                                                             (\ILLEGAL.ARG VALUE)))
                       VALUE)
                  (ENDOFSTREAMOP
                      (replace ENDOFSTREAMOP of FILE with VALUE))
                  (BUFFERS (replace MAXBUFFERS of FILE with (IMAX 1 (FIX VALUE))))
                  (CHARSET (CHARSET FILE VALUE))
                  (OR (FDEVOP 'SETFILEINFO (SETQ DEV (fetch DEVICE of FILE))
                          FILE ATTRIB VALUE DEV)
                      (SELECTQ ATTRIB
                          (LENGTH
                                 ;; Let device at this attribute first.  Probably should not have this generic op, since we don't know how to
                                 ;; do this for all devices

                                 [\SETEOFPTR FILE (COND
                                                     ((type? BYTEPTR VALUE)
                                                      VALUE)
                                                     (T (\ILLEGAL.ARG VALUE])
                          (SIZE (\SETEOFPTR FILE (UNFOLD VALUE BYTESPERPAGE)))
                          NIL]
             [(AND [SETQ DEV (\GETDEVICEFROMNAME (SETQ FULLNAME (\ADD.CONNECTED.DIR (\CONVERT-PATHNAME FILE]
```

```
                    (SETQ FULLNAME (FDEVOP 'GETFILENAME DEV FULLNAME 'OLD DEV)))
                                                        ; Name of existing file.  It's possible we should have the device
                                                        ; do recognition instead, but then we have the confusion of file
                                                        ; not found recovery in the wrong place.
              (SELECTQ ATTRIB
                  ((ACCESS OPENBYTESIZE EOLCONVENTION)
                        NIL)
                  (OR (FDEVOP 'SETFILEINFO DEV FULLNAME ATTRIB VALUE DEV)
                      (COND
                          ((EQ ATTRIB 'LENGTH)
                            (\SETCLOSEDFILELENGTH FULLNAME (COND
                                                             ((type? BYTEPTR VALUE)
                                                              VALUE)
                                                             (T (\ILLEGAL.ARG VALUE]
              (T (LISPERROR "FILE NOT FOUND" FILE]))
```

(\SETFILEPTR
```
  [LAMBDA (FILE ADR)                                                    ; Edited 11-Sep-87 16:34 by bvm:
    (LET ((STREAM (\GETSTREAM FILE)))
        [FDEVOP 'SETFILEPTR (ffetch DEVICE of STREAM)
              STREAM
              (COND
                  ((EQ ADR -1)
                   (\GETEOFPTR STREAM))
                  ((type? BYTEPTR ADR)
                   ADR)
                  (T (LISPERROR "ILLEGAL ARG" ADR]
        (if (\RUNCODED STREAM)
            then ;; always shift the character set to 0. This might be wrong sometimes, but it is more often right than wrong.  We don't do it when
                 ;; reading a non-runcoded file, since maybe the whole file is that way (unfortunately, we can't tell)

                (ACCESS-CHARSET STREAM 0))
        (freplace (STREAM CHARPOSITION) of STREAM with 0)           ; Value is not coerced!
        ADR])
```

(\BOUT16
```
  [LAMBDA (STREAM N)                                                    (* edited%: " 2-Apr-85 17:11")
    (BOUT STREAM (LRSH N 8))
    (BOUT STREAM (LOGAND N 255))
    N])
```

(\BIN16
```
  [LAMBDA (STREAM)                                                      (* edited%: " 2-Apr-85 17:11")
    (LOGOR (LLSH (BIN STREAM)
              8)
          (BIN STREAM])
)

(PUTPROPS BOUT DOPCODE (33 BOUT 0 T -1 \BOUT (4K DORADO)))
```

;; Generic functions

(DEFINEQ

(\GENERIC.BINS
```
  [LAMBDA (STREAM BASE OFF NBYTES)                                      (* bvm%: "25-MAY-83 11:41")

    ;; BINs NBYTES bytes from STREAM to memory starting at BASE+OFF.

    (FRPTQ NBYTES (\PUTBASEBYTE BASE OFF (BIN STREAM))
          (add OFF 1))
```

(\GENERIC.BOUTS
```
  [LAMBDA (STREAM BASE OFF NBYTES)                                      (* bvm%: "25-MAY-83 11:40")

    ;; BOUTs NBYTES bytes from BASE+OFF into STREAM

    (FRPTQ NBYTES (\BOUT STREAM (\GETBASEBYTE BASE OFF))
          (add OFF 1))
```

(\GENERIC.RENAMEFILE
```
  [LAMBDA (OLDDEVICE OLDFILE NEWDEVICE NEWFILE)                         ; Edited  2-Jul-90 16:03 by nm
    (if (NOT (FDEVOP 'OPENP OLDDEVICE (FULLNAME OLDFILE)
                 NIL OLDDEVICE))
        then (RESETLST
                [RESETSAVE [SETQ OLDFILE (OPENSTREAM OLDFILE 'INPUT 'OLD '((SEQUENTIAL T)
                                                                          DON'TCACHE]
                        '(AND RESETSTATE (CLOSEF? OLDVALUE]
                [COND
                    ((SETQ NEWFILE (\COPYOPENFILE OLDFILE NEWFILE))
                     (if (\DELETEFILE (CLOSEF OLDFILE))
                         then NEWFILE
                       else (CONDITIONS:RESTART-CASE (CL:ERROR 'XCL::FS-RENAMEFILE-SOURCE-COULDNT-DELETE
```

```
                                                      :PATHNAME OLDFILE)
                    (DELETE-DESTINATION NIL :CONDITION XCL::FS-RENAMEFILE-SOURCE-COULDNT-DELETE
                           :REPORT "Delete the destination file too." (DELFILE NEWFILE)
                           NIL)
                    (DONT-DELETE-DESTINATION NIL :CONDITION
                           XCL::FS-RENAMEFILE-SOURCE-COULDNT-DELETE :REPORT "Don't delete the
                           destination file.  Just returns the destination filename." NEWFILE])])
```

(**GENERIC.OPENP**
```
  [LAMBDA (FILENAME ACCESS DEVICE)                                (* hdj " 6-Oct-86 17:07")
```

;;; return all open stream on DEVICE with name FILENAME and access ACCESS. FILENAME is assumed to be fully 'recognized.' FILENAME and/or
;;; ACCESS may be NIL.

```
    (if FILENAME
        then [LET ((OPENFILES (fetch (FDEV OPENFILELST) of DEVICE)))
                   (if OPENFILES
                       then (for STREAM in OPENFILES collect STREAM when (AND (STRING-EQUAL FILENAME
                                                                                 (fetch (STREAM FULLNAME)
                                                                                    of STREAM))
                                                                 (OR (NULL ACCESS)
                                                                     (\IOMODEP STREAM ACCESS T]
        else (for S in (fetch (FDEV OPENFILELST) of DEVICE) collect S when (AND (OR (NULL ACCESS)
                                                                      (\IOMODEP S ACCESS T))
                                                                 (fetch USERVISIBLE of S]
```

(**GENERIC.READP**
```
  [LAMBDA (STREAM FLG)                                            ; Edited 11-Sep-87 16:26 by bvm:
                                                                  ; The 10 does not do the EOL check on the peeked character.
                                                                  ; If FLG is NIL, a single EOL doesn't count.
    (PROG ((SHIFTEDCHARSET (UNFOLD (ACCESS-CHARSET STREAM)
                                   256)))
          (RETURN (AND (NOT (\EOFP STREAM))
                       (OR (NOT (NULL FLG))
                           [NEQ EOL.TC (\SYNCODE \PRIMTERMSA (OR (\NSPEEK STREAM SHIFTEDCHARSET SHIFTEDCHARSET T
                                                                  )
                                                                (RETURN]
                           (UNINTERRUPTABLY
                               (\NSIN STREAM SHIFTEDCHARSET SHIFTEDCHARSET)
                               ;; To find out if the EOL is the last character, we BIN the stream, check for EOF, then back it up again.
                               (PROG1 (NOT (\EOFP STREAM))
                                      (\BACKNSCHAR STREAM SHIFTEDCHARSET)))]])
```

(**GENERIC.CHARSET**
```
  [LAMBDA (STREAM NEWVALUE)                                       ; Edited 11-Sep-87 16:20 by bvm:
```

;;; sets or returns the current numeric character set for this stream. This never writes anything on a stream, it just tells the stream what to think.

```
    (PROG1 (ffetch (STREAM CHARSET) of (\DTEST STREAM 'STREAM))
           (AND NEWVALUE (freplace (STREAM CHARSET) of STREAM with NEWVALUE)))])
)

(DEFINEQ
```

(**MAP-OPEN-STREAMS**
```
  [LAMBDA (FN DEVICES ACCESS)                                     (* hdj "11-Sep-86 10:48")
    (for DEVICE in DEVICES when (fetch (FDEV OPENP) of DEVICE)
       join (for STREAM in (FDEVOP 'OPENP DEVICE NIL ACCESS DEVICE) collect (APPLY* FN STREAM])
)

(RPAQQ FILING.TYPES ((BINARY 0)
                     (DIRECTORY 1)
                     (TEXT 2)
                     (SERIALIZED 3)
                     (INTERPRESS 4361)
                     (TEDIT 6056)
                     (FASL 6057)
                     (LAFITE 6058)))

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(GLOBALVARS FILING.TYPES)
)
```