

File created: 8-Dec-2023 15:17:12 {WMEDLEY}<sources>FILEIO.;124

edit by: rmk

changes to: (FNS \GENERIC.CHARSET CHARSET ACCESS-CHARSET)

previous date: 7-Dec-2023 23:54:02 {WMEDLEY}<sources>FILEIO.;121

Read Table: INTERLISP

Package: INTERLISP

Format: XCCS

(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)))
```

```
(MACROS TestMasked APPENDABLE APPENDONLY DIRTYABLE OPENED OVERWRITEABLE
```

```
READABLE READONLY WRITEABLE)
```

```
(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
```

```
; Device operations
```

```
(FNS \DEFINEDEVICE \GETDEVICEFROMNAME \GETDEVICEFROMHOSTNAME \REMOVEDEVICE \REMOVEDEVICE.NAMES)
```

```
(INITVARS (STREAM-AFTER-OPEN-FNS NIL))
```

```
(FNS \CLOSEFILE \DELETEFILE \DEVICEEVENT \GENERATEFILES \GENERATENEXTFILE \GENERATEFILEINFO
```

```
\GETFILENAME \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)
```

```
[INITVARS (FILING.TYPES '( (BINARY 0)
```

```
(DIRECTORY 1)
```

```
(TEXT 2)
```

```

(SERIALIZED 3)
(INTERPRESS 4361)
(TEDIT 6056)
(FASL 6057)
(LAFITE 6058]
(GLOBALVARS FILING.TYPES)
(DECLARE%: EVAL@COMPILE DONTCOPY (EXPORT (MACROS \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)
(FNS \BOUTEOL)
(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))
(LOCALVARS . T)
(DECLARE%: DONTEVAL@LOAD DOEVAL@COMPILE DONTCOPY COMPILEVAR (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 POINTER WORD WORD POINTER POINTER
POINTER POINTER POINTER)

```

:: ---field descriptor list elided by lister---

```
' 64)
```

```
(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)
(USERSVISIBLE FLAG)
(EOLCONVENTION BITS 2)
(READONLY-EXTERNALFORMAT 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)
(INCCODEFN POINTER)
(PEEKCCODEFN POINTER)
(BACKCCODEFN POINTER)
(EXTERNALFORMAT 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) ; True if stream likes to read and write more than one buffer at a
(MULTIBUFFERHINT FLAG) ; 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 CBUFSIZE unless this bit
; set and COFFSET lt 512
(CBUFDIRTY FLAG) ; true if BOUT has sullied the current buffer
(DEVICE POINTER) ; FDEV of this guy
(USERSVISIBLE FLAG) ; Listed by OPENP; NIL for terminal, dribble ...
(EOLCONVENTION BITS 2) ; End-of-line convention
(READONLY-EXTERNALFORMAT FLAG) ; T if external format can only be set at open.
; Was NOTXCCS.
(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
; Buffer chain for pmapped streams
; Max # of buffers the system will allocate.
; After READ, RATOM, etc, the charcode that will be returned (as
; a character) by LASTC. If there is none, this field is 65535.
; For use of applications programs, not devices
; Set by \EXTERNALFORMAT
(BUFFS POINTER)
(MAXBUFFERS WORD)
(LASTCCODE WORD)
(EXTRASTREAMOP POINTER)
(INCCODEFN POINTER)
(PEEKCCODEFN POINTER)
(BACKCCODEFN POINTER)
(EXTERNALFORMAT POINTER))
(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])
(SYNONYM CBUFPTR (CPPTR))
USERCLOSEABLE _ T USERVERSIBLE _ 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)
ENDOFSTREAMOP _ (FUNCTION \EOSERROR)
IMAGEOPS _ \NOIMAGEOPS EOLCONVENTION _ LF.EOLC STRMBINFN _ (FUNCTION \STREAM.NOT.OPEN)
STRMBOUTFN _ (FUNCTION \STREAM.NOT.OPEN)
LASTCCODE _ 65535 (CREATE (PROGN (\EXTERNALFORMAT DATUM (OR (FETCH (FDEV DEFAULTEXTERNALFORMAT)
OF (FFETCH (STREAM DEVICE) OF DATUM))
:DEFAULT)))
DATUM)))
)
(/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 POINTER WORD WORD POINTER POINTER
POINTER POINTER POINTER)
;; ---field descriptor list elided by lister---
' 64)
(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))
(CONSTANTS AppendBit NoBits ReadBit WriteBit (OutputBits (LOGOR AppendBit WriteBit))
  (BothBits (LOGOR ReadBit OutputBits)))
)
(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))]
)
(RPAQQ EOLCONVENTIONS ((CR.EOLC 0)
  (LF.EOLC 1)
  (CRLF.EOLC 2)
  (ANY.EOLC 3)))
(DECLARE%: EVAL@COMPILE
(RPAQQ CR.EOLC 0)
(RPAQQ LF.EOLC 1)
(RPAQQ CRLF.EOLC 2)
(RPAQQ ANY.EOLC 3)
(CONSTANTS (CR.EOLC 0)
  (LF.EOLC 1)
  (CRLF.EOLC 2)
  (ANY.EOLC 3))
)
)
;; END EXPORTED DEFINITIONS
(DEFINEQ
(STREAMPROP
[LAMBDA X
  ;; general top level entry for both fetching and setting stream properties.
  (COND
    ((IGREATERP X 2)
    (PUTSTREAMPROP (ARG X 1)
    (ARG X 2)

```

(* rda%: "22-Aug-84 14:24")

```

      (ARG X 3)))
    ((EQ X 2)
     (GETSTREAMPROP (ARG X 1)
                    (ARG X 2)))
    (T (\ILLEGAL.ARG NIL])

```

(GETSTREAMPROP

```
[LAMBDA (STREAM PROP)
```

```

; Edited 25-Aug-2023 08:45 by rmk
; Edited 5-Jul-2022 23:57 by rmk
; Edited 29-Jun-2021 17:06 by rmk:
(* rda%: "22-Aug-84 16:17")

```

```

(SELECTQ PROP
 ( (FORMAT EXTERNALFORMAT :EXTERNAL-FORMAT)
   (\EXTERNALFORMAT STREAM))
 (ENDOFSTREAMOP
  (FETCH (STREAM ENDOFSTREAMOP) OF STREAM))
 (LINELENGTH (LINELENGTH NIL STREAM))
 (LISTGET (fetch (STREAM OTHERPROPS) of STREAM)
           PROP])

```

(PUTSTREAMPROP

```
[LAMBDA (STREAM PROP VALUE)
```

```

; Edited 25-Aug-2023 08:45 by rmk
; Edited 5-Jul-2022 23:56 by rmk
; Edited 29-Jun-2021 17:06 by rmk:
(* rda%: "22-Aug-84 16:11")

```

```

(SELECTQ PROP
 ( (FORMAT EXTERNALFORMAT :EXTERNAL-FORMAT)
   ;; Return the old name (=VALUE), not the format datum. Better design: the format should have it's
   ;; name, and not have name as a separate property.
   [IF (FETCH (STREAM READONLY-EXTERNALFORMAT) OF STREAM)
       THEN (ERROR "EXTERNALFORMAT CANNOT BE CHANGED" STREAM)
       ELSE (PROG1 (\EXTERNALFORMAT STREAM NIL)
                  (AND VALUE (\EXTERNALFORMAT STREAM VALUE)))]
 (ENDOFSTREAMOP
  (PROG1 (fetch (STREAM ENDOFSTREAMOP) of STREAM)
         (replace (STREAM ENDOFSTREAMOP) of STREAM with VALUE)))
 (LINELENGTH (LINELENGTH VALUE 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)
 (AND (type? STREAM X)
      X])
)
```

```
(* rmk%: "14-OCT-83 14:35")
```

;; 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
                          14)
                          (COND
                           (NAME (NCHARS NAME))
                           (T 0))
                        (\OUTCHAR STREAM (fetch (READTABLEP HASHMACROCHAR) of *READTABLE*))
                        (\OUTCHAR STREAM (CHARCODE <))
                        (\SOUT (MKSTRING TYPENAME)

```

```
; Longest address is '</177,177777>'
```

```

    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 10-Oct-2022 15:57 by lmm
; Edited 9-Oct-2022 08:58 by lmm
; 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)))
    ((TYPE? FDEV (FETCH DEVICE OF STRM))
      ; Name the device
      (CONCAT TYPE " " [CL:STRING-CAPITALIZE
        (STRING (fetch (FDEV DEVICENAME)
          of (fetch DEVICE of STRM)
            " Stream"))
      (T (CONCAT TYPE " 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
(REMOTE P 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
; (name recog device) => full file name
; (name) => deletes file so named, returning name, or NIL on
; failure. RECOG=OLDEST
; (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
; (olddevice oldfile newdevice newfile) to rename file on this
; (olddevice) to a potentially different device.
; (name access dev) => stream if name is open for access, or all
; open streams if name = NIL
; (stream dev) => registers stream on its device
; (stream dev) => unregisters a stream from its device
; (host/dir dev) => # of free pages on host/dir
; (host/dir dev)
; (name dev) => name if it is well-formed file name for dev
; (host dev) => true if host is alive, i.e., responsive; only defined
; if REMOTEP is true
; (host fastp dev) => closes connections to host

; -----The following are required methods for operating on open streams-----
; (BIN POINTER) ; (stream) => next byte of input
; (BOUNT POINTER) ; (stream byte) output byte to stream
; (PEEKBIN POINTER) ; (stream) => next byte without advancing position in stream
; (FDEV1 POINTER) ; Was READCHAR, replaced by READCHARCODE. Now
; available for device-specific use
; (FDEV2 POINTER) ; Was WRITECHAR (stream char) => writes char to stream
; (FDEV3 POINTER) ; Was PEEKCHAR
; (FDEV4 POINTER) ; Was UNREADCHAR
; (READP POINTER) ; (stream flag) => T if there is input available from stream right
; now
; (stream) => T if BIN would signal eof.
; (stream buffer byteoffset nbytes)
; (stream buffer byteoffset nbytes)
; (stream waitforFinish) flushes out to device anything that is
; buffered awaiting transmission
; (stream/name attribute device) => value of attribute for open
; stream or name of closed file
; (stream/name attribute newvalue device) sets attribute of open
; stream or closed file of given name
; (stream charset) => access function for the charset slot, for
; benefit of indirect streams. See IMCHARSET for changing it on
; a file.
; (stream) => indirected input stream
; (stream) => indirected output stream

; -----Following are required of random-access streams-----
; (GETFILEPTR POINTER) ; (stream) backs up 'fileptr' by one. Stream is only required to be
; (GETEOFPTR POINTER) ; able to do this once, i.e. one-character buffer suffices
; (SETFILEPTR POINTER) ; (stream length) => truncates or lengthens stream to indicated
; (BACKFILEPTR POINTER) ; length
; ; Should be possible only if RANDOMACCESSP

; (SETEOFPTR POINTER)

; (LASTC POINTER)

; -----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)-----
; (DEFAULTEXTERNALFORMAT POINTER) ; Was READCHARCODE. Read a character code from the
; ; stream (cf BIN for bytes).

)
DIRECTORYNAMEP _ (FUNCTION NIL)
HOSTNAMEP _ (FUNCTION NIL)
READP _ (FUNCTION \GENERIC.READP)
SETFILEPTR _ (FUNCTION \IS.NOT.RANDACCESSP)
GETFILEPTR _ (FUNCTION \ILLEGAL.DEVICEOP)
GETEOFPTR _ (FUNCTION \IS.NOT.RANDACCESSP)

```



```

(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)
(DEFAULTTEXTINTERNALFORMAT POINTER)))

```

:: 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 (CDR 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]))
```

(RPAQ? STREAM-AFTER-OPEN-FNS NIL)

(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 DEPTH)
  ;; Edited 29-Mar-2022 08:52 by rmk: Added local DEPTH parameter, defaults to the free FILING.ENUMERATION.DEPTH.
  ;; 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))
  (LET ((FDEV (\GETDEVICEFROMNAME PATTERN))
        (FILING.ENUMERATION.DEPTH (IF (FIXP DEPTH)
                                       ELSEIF DEPTH
                                       THEN MAX.SMALLP
                                       ELSE FILING.ENUMERATION.DEPTH)))
    (DECLARE (SPECVARS FILING.ENUMERATION.DEPTH))
    (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.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 25-Aug-2023 08:43 by rmk
; Edited 6-Jul-2022 00:00 by rmk
; Edited 19-Dec-2021 09:30 by rmk
; Edited 14-Dec-2021 16:10 by rmk
; Edited 13-Dec-2021 15:20 by rmk
; Edited 29-Jun-2021 17:07 by rmk:
; Edited 5-Oct-92 13:45 by jds

```

```

;; Does generic parameters when a file/stream is open. Called by \OPENFILE and OPENSTREAM
;; RMK: August 2023: Added PUTSTREAMPROP as last resort.
;; RMK July 2020: Make sure that \EXTERNALFORMAT is always called, so that it can implement per-device defaults.
;; RMK August 2020: Added hook for user STREAM-AFTER-OPEN-FNS, not global so can be rebound.

```

```

(DECLARE (USEDFFREE STREAM-AFTER-OPEN-FNS))
(\EXTERNALFORMAT STREAM :DEFAULT)
(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))
  ((FORMAT EXTERNALFORMAT :EXTERNAL-FORMAT)
   (\EXTERNALFORMAT STREAM VAL))
  (CONVHANKAKU (CONVHANKAKU STREAM VAL))
  ((EOL EOLCONVENTION EOLC)
   (SETFILEINFO STREAM 'EOL VAL))
  (PUTSTREAMPROP STREAM ATTR VAL)))
(FOR FN IN STREAM-AFTER-OPEN-FNS DO (APPLY* FN STREAM ACCESS PARAMETERS))

```

(\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.REVALIDATEFILELIST

[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 PMAPEd 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)
  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) (* hjd "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)
     NIL)
    (T (replace F1 of STREAM with (fetch F1 of NEWSTREAM))
       (replace F2 of STREAM with (fetch F2 of NEWSTREAM))
       (replace F3 of STREAM with (fetch F3 of NEWSTREAM))
       (replace F4 of STREAM with (fetch F4 of NEWSTREAM))
       (replace F5 of STREAM with (fetch F5 of NEWSTREAM))
       (replace FW6 of STREAM with (fetch FW6 of NEWSTREAM))
       (replace FW7 of STREAM with (fetch FW7 of NEWSTREAM))
       (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])
       ; Nothing changed
       ; Copy 'device' information from the new opening to the old
```

(\BUFFERED.REVALIDATEFILELST

[LAMBDA (DEVICE) (* hjd "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)
     [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])
     ; the file changed or disappeared, so zap the stream
     ; might as well return something useful
     (FDEVOP 'OPENP DEVICE NIL NIL DEVICE))])
```

(\PRINT-REVALIDATION-RESULT

[LAMBDA (RESULT STREAM) ; Edited 29-Sep-2022 20:11 by Imm (* hjd "26-May-86 15:46")

:: stack overflow if DRIBBLEFILE; use PROMPTWINDOW

```
(FRESHLINE PROMPTWINDOW)
(if [AND (DRIBBLEFILE)
  (NOT (OPENP (DRIBBLEFILE)
              'APPEND])
  THEN (PRINTOUT PROMPTWINDOW "Dribble file " (DRIBBLE)
    " ended" T))
  (printout PROMPTWINDOW "**** WARNING: The file " (fetch (STREAM FULLNAME) of STREAM)
  (SELECTQ RESULT
    (CHANGED " has been modified since you last accessed it!")
    (DELETED " was previously opened but has disappeared!")
    (SHOULDNT))
  T])
```


(\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 NIL])
```

(\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)
EOF _ (\INHERITFDEVOP.S EOF 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 20-May-92 11:08 by jds

;; 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
 (SETQ DN \CONNECTED.DIRECTORY))
 (NIL (SETQ DN (OR LOGINHOST/DIR '{DSK})))
 (COND
 [(AND [SETQ FDEV
 (LET [(HOST (FILENAMEFIELD DIRNAME 'HOST)
 (SELCHARQ (NTHCHARCODE DIRNAME 1)
 (>
 ; Remove leading > from a subdirectory spec.
 (SETQ DIRNAME (SUBSTRING DIRNAME 2)))
 NIL)
 (\bGETDEVICEFROMHOSTNAME
 (OR HOST (FILENAMEFIELD [SELCHARQ (NTHCHARCODE DIRNAME 1)
 ((< /)
 ; Whole directory, use it all.
 (SETQ DIRNAME (PACKFILENAME.STRING
 'DIRECTORY DIRNAME
 'BODY
 \CONNECTED.DIRECTORY)))
 (SELCHARQ (NTHCHARCODE DIRNAME (NCHARS
 DIRNAME
))
 ((> /)
 ; Remove any trailing > or / from a subdirectory spec.
 (SETQ DIRNAME
 (PACKFILENAME.STRING
 'SUBDIRECTORY
 (SUBSTRING DIRNAME 1 -2)
 'DIRECTORY
 \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 (COND
 ((NOT STRPTR)
 (MKSTRING DN))
 ((EQ STRPTR T)
 (MKATOM DN))
 (T (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 (\bGETDEVICEFROMNAME 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) (* Imm "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 [fetch 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 [fetch 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
(START) ; 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 (\EOF SRC) do (\BOUT DST (\BIN SRC]
(RETURN T) ; As specified in VM
)])
)])
```

(COPYCHARS

```
[LAMBDA (SRCFIL DSTFIL START END) ; Edited 13-Aug-2021 18:39 by rmk:
; Edited 14-Jun-2021 22:08 by rmk:
; Edited 8-Dec-95 16:38 by rmk:
; Edited 26-Mar-99 12:13 by rmk:
;; This is similar to COPYBYTES except that conversion is done between the EOL convention and externalformat of the input and the EOL
;; convention/external format of the output
[PROG ((SRCSTRM (\GETSTREAM SRCFIL))
(DSTSTRM (\GETSTREAM DSTFIL))
(ACTUALSTART 0)
RAP ACTUALEND EOF SRCEOLC DSTEOLC CH)
(CL:WHEN (AND (EQ (SETQ SRCEOLC (fetch EOLCONVENTION of SRCSTRM))
(SETQ DSTEOLC (fetch EOLCONVENTION of DSTSTRM)))
(EQ (FETCH EXTERNALFORMAT OF SRCSTRM)
(FETCH EXTERNALFORMAT OF DSTSTRM)))
(RETURN (COPYBYTES SRCSTRM DSTSTRM START END)))
;; Format or EOL convention are different. So first decode the START END specification
[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
;; Not random access and START and END are both NIL, just copy to the end of file,no need to count.
(until (\EOFP SRCSTRM) do (\OUTCHAR DSTSTRM (\INCCODE.EOLC SRCSTRM SRCEOLC))
(RETURN))
(CL:UNLESS (IGEQ ACTUALEND ACTUALSTART)
 (ERROR "Negative number of bytes to copy" (IDIFFERENCE ACTUALEND ACTUALSTART)))
;; We now know which bytes we need to copy, in the case that there is an EOL/format mismatch. If we assume that this is fairly unusual and that
;; we don't want to assume here that we know how the CR and LF are byte-coded, we don't try to optimize for an EOL-only change. We just go
;; generic.
;; The \INCCODE.EOLC and \OUTCHAR handle all format and EOL issues.
(BIND (CNT _ (IDIFFERENCE ACTUALEND ACTUALSTART)) DECLARE (SPECVARS CNT)
 WHILE (IGREATERP CNT 0) DO (\OUTCHAR DSTSTRM (\INCCODE.EOLC SRCSTRM NIL 'CNT CNT]
T])

```

(COPYFILE

```

[LAMBDA (FROMFILE TOFILE DESTPARAMETERS SOURCEPARAMETERS)
;; Edited 8-Jul-2022 10:54 by rmk: Added SOURCEPARAMETERS, in particular to declare external format
;; Edited 8-Jul-2022 10:41 by rmk
;; 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)
 ,@SOURCEPARAMETERS]
' (PROGN (CLOSEF OLDVALUE]
(\COPYOPENFILE FROMFILE TOFILE DESTPARAMETERS)))]

```

(\COPYOPENFILE

```

[LAMBDA (INSTREAM NEWNAME DESTPARAMETERS)
;; Edited 8-Jul-2022 10:58 by rmk: Use COPYCHARS if external formats are different
;; Edited 3-May-2021 20:36 by rmk:
;; Edited 11-Dec-95 11:50 by
;; 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 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.

```

(CL:UNLESS (EQ TYPE 'TEXT)

;; RMK: Setting the LENGTH seems wrong for TEXT files, since the byte-length could change if EOL or external-format differs. Let

;; normal Length mechanisms prevail. Indeed, why bother with setting the length anyway--unless this is merely a hint for the opener?

;; If so, the text guard can be removed.

```

[COND
  ((SETQ X (GETFILEINFO INSTREAM 'LENGTH))
(push PROPS (LIST 'LENGTH X])
[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
  ((OR (EQ TYPE 'TEXT)
        (NEQ (fetch (STREAM EXTERNALFORMAT) of INSTREAM)
              (fetch (STREAM EXTERNALFORMAT) of OUTSTREAM))))
```

:: RMK: COPYCHARS ensures that external format conversion happens if necessary

```
(COPYCHARS INSTREAM OUTSTREAM)
(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 NIL))
          [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 DONTMARKFILE) ; Edited 8-Dec-2023 15:04 by rmk
 ; 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,
 ;; depending on the external format, unless DONTMARKFILE). ACCESS-CHARSET recurses through any commonlisp meta-streams, eventually
 ;; reaches \GENERIC.CHARSET, which then applies the format's FORMATCHARSETFN.

```
(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)
      (if (EQ OLDVAL NSCHARSETSHIFT)
          then (SETQ OLDVAL T)
          OLDVAL])

```

; First modify the stream's slot

(ACCESS-CHARSET

```

[LAMBDA (STREAM NEWVALUE DONTMARKFILE)
  ;; Unless DONTMARKSTREAM, if STREAM is open for output, the external format function may modify the backing file as well as the stream, e.g.
  ;; put in XCCS shifting bytes.
  (FDEVOP 'CHARSETFN (fetch (STREAM DEVICE) of STREAM)
    STREAM NEWVALUE DONTMARKFILE])

```

; Edited 8-Dec-2023 15:05 by rmk
; Edited 11-Sep-87 15:46 by bvm:

(GETEOFPTR

```

[LAMBDA (FILE)
  (PROG ((STREAM (\GETSTREAM FILE)))
    (RETURN (FDEVOP 'GETEOFPTR (fetch DEVICE of STREAM)
      STREAM]))

```

(* rmk%: "21-OCT-83 11:19")

(GETFILEINFO

```

[LAMBDA (FILE ATTRIB)
  (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)
                (ANY.EOLC 'ANY)
                (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))]
              ((FORMAT EXTERNALFORMAT)
                (\EXTERNALFORMAT 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)
  (OR (CADR (ASSOC TYPE FILING.TYPES))
      (FIXP TYPE]))
```

(* bvm%: "15-Jan-85 17:08")

(GETFILEPTR

```
[LAMBDA (FILE)
  (PROG ((STREAM (\GETSTREAM FILE)))
    (RETURN (FDEVOP 'GETFILEPTR (fetch DEVICE of STREAM)
      STREAM))
```

(* rmk%: "21-OCT-83 11:19")

(SETFILEINFO

```
[LAMBDA (FILE ATTRIB VALUE)
```

; Edited 19-Dec-2021 09:30 by rmk
; Edited 29-Jun-2021 17:05 by rmk:
; Edited 11-Dec-95 11:08 by
; 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)
  NIL)
  (EOL (replace EOLCONVENTION of FILE with (SELECTQ VALUE
```

; These can't be changed for an open file

```
(CR CR.EOLC)
(CRLF CRLF.EOLC)
(LF LF.EOLC)
(ANY (CL:WHEN (\GETSTREAM FILE 'OUTPUT T)
  (ERROR "EOL convention ANY is not
  allowed for output streams" FILE
  ))
  ANY.EOLC)
(\ILLEGAL.ARG VALUE)))
```

```
VALUE)
((FORMAT EXTERNALFORMAT)
  (\EXTERNALFORMAT FILE 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

```
[\SETEOPTR FILE (COND
  ((type? BYTEPTR VALUE)
  VALUE)
  (T (\ILLEGAL.ARG VALUE]))
  (SIZE (\SETEOPTR 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)
  (LET ((STREAM (\GETSTREAM FILE)))
    [FDEVOP 'SETFILEPTR (ffetch DEVICE of STREAM)
      STREAM
      (COND
        ((EQ ADR -1)
          (\GETEOPTR STREAM))
        ((type? BYTEPTR ADR)
          ADR)
        (T (LISPERROR "ILLEGAL ARG" ADR)
```

; Edited 21-Jun-2021 12:12 by rmk:

;; RMK: There is no reason to believe that going to CSET 0 is more often right than wrong. If it truly is a runcoded cset 0 file, where this
;; would be appropriate, then it presumably is already in cset 0, no need to do anything. If it is runcoded in some other character set (e.g.
;; Greek), then it is more likely that the whole file (or at least wherever we are setting the file ptr) is also in Greek. So leave it alone.

:: And this would only apply, presumably, to an NS/XCCS file or some other, unencoded file format, of which there aren't any.

(* 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)))
; Value is not coerced!

(**replace** (STREAM CHARPOSITION) **of** STREAM **with** 0)
ADR])

(BOUT16

[LAMBDA (STREAM N)
(**BOUT** STREAM (LRSH N 8))
(**BOUT** STREAM (LOGAND N 255))
N])

(* edited%: " 2-Apr-85 17:11")

(BIN16

[LAMBDA (STREAM)
(LOGOR (LLSH (**BIN** STREAM)
8)
(**BIN** STREAM)])

(* edited%: " 2-Apr-85 17:11")

)

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

:: Generic functions

(DEFINEQ

(\GENERIC.BINS

[LAMBDA (STREAM BASE OFF NBYTES)
;; BINS NBYTES bytes from STREAM to memory starting at BASE+OFF.
(FRPTQ NBYTES (\PUTBASEBYTE BASE OFF (\bIN STREAM))
(**add** OFF 1])

(* bvm%: "25-MAY-83 11:41")

(\GENERIC.BOUTS

[LAMBDA (STREAM BASE OFF NBYTES)
;; BOUTs NBYTES bytes from BASE+OFF into STREAM
(FRPTQ NBYTES (\bOUT STREAM (\GETBASEBYTE BASE OFF))
(**add** OFF 1])

(* bvm%: "25-MAY-83 11:40")

(\GENERIC.RENAMEFILE

[LAMBDA (OLDDEVICE OLDFILE NEWDEVICE NEWFILE)
(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]]))

; Edited 2-Jul-90 16:03 by nm

(\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** USERVERSIBLE **of** S])

(\GENERIC.READP

[LAMBDA (STRM FLG)

; Edited 19-Jul-2022 23:23 by rmk
 ; Edited 23-Jun-2021 13:09 by rmk:
 ; The 10 does not do the EOL check on the peeked character.
 ; If FLG is NIL, a single EOL doesn't count.

```
(CL:UNLESS (\EOFP STRM)
  [PROG NIL
    (RETURN (OR FLG [NEQ EOL.TC (\SYNCCODE \PRIMTERMSA (OR (\PEEKCCODE.EOLC STRM T)
      (RETURN]
        (UNINTERRUPTABLY
          (\INCCODE.EOLC STRM)
          (PROG1 (NOT (\EOFP STRM))
            (\BACKCCODE.EOLC STRM))))))])])])
```

; Read what we peeked (EOLC), see if anything comes after

(\GENERIC.CHARSET

[LAMBDA (STREAM NEWVALUE DONTMARKSTREAM)

; Edited 8-Dec-2023 15:17 by rmk
 ; Edited 11-Sep-87 16:20 by bvm:

;;; sets or returns the current numeric character set for this stream. This applies the stream's FORMATCHARSETFN if it has one, and (if
 ;;; MARKSTREAM) that may change an output backing stream in some way (e.g. write XCCS charset shift bytes). Otherwise, this just sets the charset
 ;;; stream parameter to influence subsequent reading and writing behavior. Charset doesn't exist in some formats (e.g. UTF-8), the format function
 ;;; would be a noop in that case.

```
(\DTEST STREAM 'STREAM)
(LET ((EFORMAT (ffetch (STREAM EXTERNALFORMAT) of STREAM))
      OLDVALUE)
  (if (AND EFORMAT (fetch (EXTERNALFORMAT FORMATCHARSETFN) of EFORMAT))
      then (APPLY* (fetch (EXTERNALFORMAT FORMATCHARSETFN) of EFORMAT)
                  STREAM NEWVALUE DONTMARKSTREAM)
      else (SETQ OLDVALUE (ffetch (STREAM CHARSET) of STREAM))
          (CL:WHEN NEWVALUE
            (freplace (STREAM CHARSET) of STREAM with NEWVALUE)
            (CL:UNLESS (OR DONTMARKSTREAM (EQ NEWVALUE OLDVALUE)
              (NOT (\IOMODEP STREAM 'OUTPUT T)))
              (\BOUT STREAM NSCHARSETSHIFT)
              (if (OR (EQ CHARSET T)
                (EQ CHARSET NSCHARSETSHIFT))
                  then (\BOUT STREAM NSCHARSETSHIFT)
                  (\BOUT STREAM 0)
                  else (\BOUT STREAM CHARSET))))
            OLDVALUE]))
```

)

(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])
```

)

```
(RPAQ? 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)

)

(DECLARE%: EVAL@COMPILE DONTCOPY

;; FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

```
(PUTPROPS \DEVICE-OPEN-STREAMS MACRO [ARGS (LET ((DEVICE (CAR ARGS)))
  `(FDEVOP 'OPENP ,DEVICE NIL NIL ,DEVICE)])
```

(PUTPROPS \CONVERT-PATHNAME DMACRO (OPENLAMBDA (PATHNAME?)

;; Coerce pathnames to Interlisp strings, for the benefit of antediluvian Interlisp-D file fns

```
(CL:TYPECASE PATHNAME?
  (PATHNAME (INTERLISP-NAMESTRING PATHNAME?))
  (T PATHNAME?)))
```

)

```
(DEFOPTIMIZER ACCESS-CHARSET (STREAM &OPTIONAL NEWVALUE MARKSTREAM)
  `((OPENLAMBDA (STRM)
    (FDEVOP 'CHARSETFN (fetch (STREAM DEVICE) of STRM)
      STRM
      ,NEWVALUE
      ,MARKSTREAM)
    ,STREAM))
)
```

:: END EXPORTED DEFINITIONS

(DECLARE%: DONTEVAL@LOAD DOCOPY

```
[MAPC ' ( (FORCEOUTPUT FLUSHOUTPUT)
  (FORCEOUTPUT FLUSHMAP)
  (\GENERIC.BINS \NONPAGEDBINS)
  (\GENERIC.BOUTS \NONPAGEDBOUTS))
(FUNCTION (LAMBDA (PAIR)
  (PUTD (CADR PAIR)
    (GETD (CAR PAIR))
    T])
)
```

:: Internal functions

(DEFINEQ

```
(\EOF.ACTION
  [LAMBDA (STREAM)
    ;; Standard thing to do at end of stream
    (CL:FUNCALL (fetch (STREAM ENDOFSTREAMOP) of STREAM)
      STREAM])
  (* bvm%: "24-Aug-84 18:06")
```

```
(\EOSERROR
  [LAMBDA (STREAM)
    (LISPERROR "END OF FILE" (fetch (STREAM FULLNAME) of STREAM)
      T])
  (* hdj "17-Jun-86 18:35")
```

```
(\GETEOFPTR
  [LAMBDA (STREAM)
    (FDEVOP 'GETEOFPTR (fetch DEVICE of STREAM)
      STREAM])
  (* Imm "25-MAY-83 23:17")
```

```
(\INCFILEPTR
  [LAMBDA (STREAM AMOUNT)
    (COND
      ((NOT (fetch PAGEMAPPED of (fetch DEVICE of STREAM)))
        (\SETFILEPTR STREAM (IPLUS (\GETFILEPTR STREAM)
          AMOUNT)))
      (T (\PAGED.INCFILEPTR STREAM AMOUNT]))
  (* bvm%: " 7-Jun-84 16:47")
```

```
(\PEEKBIN
  [LAMBDA (STREAM NOERRORFLG)
    (FDEVOP 'PEEKBIN (fetch DEVICE of STREAM)
      STREAM NOERRORFLG])
  (* bvm%: "26-DEC-81 15:59")
```

```
(\SETCLOSEDFILELENGTH
  [LAMBDA (FILENAME NBYTES)
    ;; Reset the length of a closed file to nBytes.
    (PROG [(STREAM (\OPENFILE FILENAME 'BOTH 'OLD)
      (\SETEOFPTR STREAM NBYTES)
      (\CLOSEFILE STREAM)
      (RETURN T])
  (* bvm%: "13-JUL-83 15:15")
```

```
(\SETEOFPTR
  [LAMBDA (STREAM LEN)
    (FDEVOP 'SETEOFPTR (fetch DEVICE of STREAM)
      STREAM LEN])
  (* bvm%: " 9-Jul-84 17:37")
```

```
(\SETFILEPTR
  [LAMBDA (STREAM INDX)
    ;; Fast case of SETFILEPTR, assumes STREAM is a stream and INDX is an already coerced fileptr (not -1) Does not reset CHARPOSITION and
    ;; value is uninteresting
    (FDEVOP 'SETFILEPTR (fetch DEVICE of STREAM)
      STREAM INDX])
  (* rmk%: "22-AUG-83 13:37")
```

)

(DEFINEQ

(\FIXPOUT

```
[LAMBDA (STRM N) (* rmk%: "25-Jun-84 14:47")
  (\BOUT STRM (LOADBYTE N 24 BITSPERBYTE))
  (\BOUT STRM (LOADBYTE N 16 BITSPERBYTE))
  (\BOUT STRM (LOADBYTE N 8 BITSPERBYTE))
  (\BOUT STRM (LOADBYTE N 0 BITSPERBYTE])
```

(\FIXPIN

```
[LAMBDA (STRM) (* rmk%: "14-Jun-84 19:36")
  ;; Read in a full 32 bit integer
  (LOGOR (LLSH (\WIN STRM
    16)
    (\WIN STRM])
```

)

(DEFINEQ

(\BOUTEOL

```
[LAMBDA (STREAM) ; Edited 6-Aug-2021 14:51 by rmk:
  ;; Convenient closed function to put out EOL characters without depending on EXPORTS.ALL for constants. .
  (SELECTC (FETCH (STREAM EOLCONVENTION) OF STREAM)
    (LF.EOLC (\BOUT STREAM (CHARCODE LF)))
    (CR.EOLC (\BOUT STREAM (CHARCODE CR)))
    (CRLF.EOLC (\BOUT STREAM (CHARCODE CR))
      (\BOUT STREAM (CHARCODE LF)))
    (ANY.EOLC (SHOULDNT))
    NIL])
```

)

(DECLARE%: DONTCOPY

:: FOLLOWING DEFINITIONS EXPORTED

(DECLARE%: EVAL@COMPILE

```
(PUTPROPS \DECFILEPTR MACRO ((STREAM X)
  (\INCFILEPTR STREAM (IMINUS X))))
```

```
(PUTPROPS \GETFILEPTR MACRO (OPENLAMBDA (STRM)
  (FDEVOP 'GETFILEPTR (fetch DEVICE of STRM)
    STRM)))
```

```
(PUTPROPS \SIGNEDWIN MACRO ((STREAM)
  (SIGNED (\WIN STREAM)
    BITSPERWORD)))
```

```
(PUTPROPS \SIGNEDWOUT MACRO ((STREAM N)
  (\WOUT STREAM (UNSIGNED N BITSPERWORD))))
```

```
(PUTPROPS \WIN MACRO (OPENLAMBDA (STREAM)
  (create WORD
    HIBYTE _ (\BIN STREAM)
    LOBYTE _ (\BIN STREAM))))
```

```
(PUTPROPS \WOUT MACRO (OPENLAMBDA (STREAM W)
  (\BOUT STREAM (fetch HIBYTE of W))
  (\BOUT STREAM (fetch LOBYTE of W))))
```

```
(PUTPROPS \BINS BYTEMACRO (OPENLAMBDA (STRM BASE OFF NBYTES)
  (FDEVOP 'BLOCKIN (fetch (STREAM DEVICE) of STRM)
    STRM BASE OFF NBYTES)))
```

```
(PUTPROPS \BOUTS BYTEMACRO (OPENLAMBDA (STRM BASE OFF NBYTES)
  (FDEVOP 'BLOCKOUT (fetch (STREAM DEVICE) of STRM)
    STRM BASE OFF NBYTES)))
```

```
(PUTPROPS \EOFP BYTEMACRO (OPENLAMBDA (STRM)
  (FDEVOP 'EOFP (fetch (STREAM DEVICE) of STRM)
    STRM)))
```

```
(PUTPROPS \SIZE.FROM.LENGTH MACRO [LAMBDA (LEN)
  (DECLARE (LOCALVARS LEN))
  (AND LEN (FOLDHI LEN BYTESPERPAGE])
```

)

(DECLARE%: EVAL@COMPILE

```

(RPAQQ BitsPerByte 8)

(RPAQ ByteOffsetSize (SELECTQ (SYSTEMTYPE)
                               (VAX 10)
                               9))

(RPAQQ WordsPerPage 256)

(CONSTANTS BitsPerByte (ByteOffsetSize (SELECTQ (SYSTEMTYPE)
                                                  (VAX 10)
                                                  9))
          WordsPerPage)
)

(DECLARE%: EVAL@COMPILE

(RPAQ \MAXFILEPTR (SUB1 (LLSH 1 30)))

[CONSTANTS (\MAXFILEPTR (SUB1 (LLSH 1 30)
)

(DECLARE%: EVAL@COMPILE

(ACCESSFNS BYTEPTR ((PAGE (FOLDLO DATUM BYTESPERPAGE))
                      (OFFSET (MOD DATUM BYTESPERPAGE)))
  (TYPE? (AND (FIXP DATUM)
              (IGEQ DATUM 0)
              (ILEQ DATUM \MAXFILEPTR)))
  (CREATE (IPLUS (UNFOLD PAGE BYTESPERPAGE)
              OFFSET)))
)

```

:: END EXPORTED DEFINITIONS

```

(DECLARE%: EVAL@COMPILE

(RPAQQ MaxChar 255)

(CONSTANTS MaxChar)
)

```

:: Buffered IO

```

(DEFINEQ

(BUFFERED.BIN
  [LAMBDA (STREAM)
    (PROG (OFF X)
      RETRY
        [RETURN (\GETBASEBYTE (OR (fetch CBUFPTR of STREAM)
                                (GO REFILL))
          (PROG1 (SETQ OFF (fetch COFFSET of STREAM))
            (COND
              ((IGEQ OFF (fetch CBUFSIZE of STREAM))
                (GO REFILL)))
            (replace COFFSET of STREAM with (ADD1 OFF)))]
        REFILL
          (COND
            ((EQ (SETQ X (FDEVOP 'GETNEXTBUFFER (fetch DEVICE of STREAM)
              STREAM
              'READ))
              T)
              (GO RETRY))
            (T (RETURN X]))

```

(* bvm%: "10-Jul-84 13:25")

```

(BUFFERED.PEEKBIN
  [LAMBDA (STREAM NOERRORFLG)
    (PROG (OFF X)
      RETRY
        [RETURN (\GETBASEBYTE (OR (fetch CBUFPTR of STREAM)
                                (GO REFILL))
          (PROG1 (SETQ OFF (fetch COFFSET of STREAM))
            (COND
              ((IGEQ OFF (fetch CBUFSIZE of STREAM))
                (GO REFILL)))
            (GO REFILL)))]
        REFILL
          (COND
            ((EQ (SETQ X (FDEVOP 'GETNEXTBUFFER (fetch DEVICE of STREAM)
              STREAM
              'READ NOERRORFLG))
              T)
              (GO RETRY))
            (T (RETURN X]))

```

(* bvm%: "24-Aug-84 17:43")

(\BUFFERED.BOUT

```
[LAMBDA (STREAM BYTE) (* bvm%: "10-Jul-84 13:30")
(CHECK (type? STREAM STREAM)
(WRITEABLE STREAM))
(PROG (OFF)
RETRY
(\PUTBASEBYTE (OR (fetch CBUFPTR of STREAM)
(GO REFILL))
(PROG1 (SETQ OFF (fetch COFFSET of STREAM))
(COND
((ILESSP OFF (fetch CBUFMAXSIZE of STREAM))
(replace COFFSET of STREAM with (ADD1 OFF)))
(T (GO REFILL))))
BYTE)
(replace CBUFDIRTY of STREAM with T)
(RETURN 1)
REFILL
(FDEVOP 'GETNEXTBUFFER (fetch DEVICE of STREAM)
STREAM
'WRITE)
(GO RETRY])
```

(\BUFFERED.BINS

```
[LAMBDA (STREAM DBASE OFFSET NBYTES NOERRORFLG) (* bvm%: "11-Jul-84 19:15")
```

;;; For buffered streams, BINs NBYTES bytes from STREAM to memory starting at DBASE+OFFSET --- If NOERRORFLG then stops without error at eof. Returns number of bytes actually read

```
(bind (BYTESLEFT _ NBYTES)
CNT END IBASE START X do [COND
((SETQ IBASE (fetch CBUFPTR of STREAM))
; Current buffer
(SETQ START (fetch COFFSET of STREAM))
; Offset of first byte to transfer
[COND
((IGREATERP (SETQ CNT (IDIFFERENCE (SETQ END (fetch CBUFSIZE
of STREAM))
START))
BYTESLEFT) ; Not a whole buffer full
(SETQ END (IPLUS START (SETQ CNT BYTESLEFT)
; First byte BEYOND whats to be read from this page
(\MOVEBYTES IBASE START DBASE OFFSET CNT)
(replace COFFSET of STREAM with END)
(COND
((EQ CNT BYTESLEFT) ; Finished
(RETURN NBYTES))
(T (add OFFSET CNT)
(SETQ BYTESLEFT (IDIFFERENCE BYTESLEFT CNT)]
(COND
((NULL (SETQ X (FDEVOP 'GETNEXTBUFFER (fetch DEVICE of STREAM)
STREAM
' READ NOERRORFLG))
; No error at eof
(RETURN (IDIFFERENCE NBYTES BYTESLEFT))
((NEQ X T) ; At eof, but EOF op returned a value to fake more data at eof
(RETURN (do (\PUTBASEBYTE DBASE OFFSET X)
(add OFFSET 1)
(COND
((EQ (add BYTESLEFT -1)
0)
(RETURN NBYTES)))
(SETQ X (\BIN STREAM])
```

(\BUFFERED.BOUTS

```
[LAMBDA (STREAM SBASE OFFSET NBYTES) (* bvm%: "10-Jul-84 13:39")
```

;;; For buffered streams, bouts NBYTES bytes to STREAM from SBASE+OFFSET

```
(bind (DEV _ (fetch DEVICE of STREAM))
CNT END DBASE START do [COND
((SETQ DBASE (fetch CBUFPTR of STREAM))
(SETQ START (fetch COFFSET of STREAM))
[COND
((IGREATERP (SETQ CNT (IDIFFERENCE (SETQ END (fetch CBUFMAXSIZE
of STREAM))
START))
NBYTES)
(SETQ END (IPLUS START (SETQ CNT NBYTES)
(\MOVEBYTES SBASE OFFSET DBASE START CNT)
(replace COFFSET of STREAM with END)
(replace CBUFDIRTY of STREAM with T)
(COND
((ILEQ (SETQ NBYTES (IDIFFERENCE NBYTES CNT))
0)
```



```

      (RETURN)
      (T (add OFFSET CNT]
(FDEVOP 'GETNEXTBUFFER DEV STREAM 'WRITE])

```

(\BUFFERED.COPYBYTES

[LAMBDA (SRC DST NBYTES)

(* bvm%: "10-Jul-84 21:48")

::: Copies NBYTES bytes from buffered stream SRC to arbitrary stream DST, or copies to eof if NBYTES is NIL

```

(bind (NOERRORFLG _ (NULL NBYTES))
      (DEV _ (fetch DEVICE of SRC))
      BUF NB STARTOFFSET END do [COND
        ((SETQ BUF (fetch CBUFPTR of SRC))
         ; Copy a buffer full
         [SETQ NB (IDIFFERENCE (SETQ END (fetch CBUFSIZE of SRC))
                               (SETQ STARTOFFSET (fetch COFFSET of SRC))
          [COND
            ((AND NBYTES (IGREATERP NB NBYTES))
             ; Don't copy too much
             (SETQ END (IPLUS STARTOFFSET (SETQ NB NBYTES))
            (\BOUTS DST BUF STARTOFFSET NB)
            (replace COFFSET of SRC with END)
            (COND
              (NBYTES (COND
                ((EQ NB NBYTES)
                 (RETURN))
                (T (SETQ NBYTES (IDIFFERENCE NBYTES NB]

```

```

repeatwhile (FDEVOP 'GETNEXTBUFFER DEV SRC 'READ NOERRORFLG])

```

)

(DECLARE%: DOEVAL@COMPILE DONTCOPY

(LOCALVARS . T)

)

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

(ADDTOVAR NLAMA)

(ADDTOVAR NLAML)

(ADDTOVAR LAMA \IS.NOT.RANDACCESSP \ILLEGAL.DEVICEOP STREAMPROP)

)

FUNCTION INDEX

ACCESS-CHARSET25 \BUFFERED.BINS32 \GENERIC.READP28
BIN21 \BUFFERED.BOUT32 \GENERIC.RENAMEFILE27
BIN1627 \BUFFERED.BOUTS32 \GETACCESS7
BOUT21 \BUFFERED.COPYBYTES33 \GETDEVICEFROMHOSTNAME11
BOUT1627 \BUFFERED.PEEKBIN31 \GETDEVICEFROMNAME11
CHARSET24 \BUFFERED.REVALIDATEFILE16 \GETEOFPTR29
CNDIR19 \BUFFERED.REVALIDATEFILELST16 \GETFILENAME13
COPYBYTES22 \CLOSEFILE12 \ILLEGAL.DEVICEOP18
COPYCHARS22 \COPYOPENFILE23 \INCFILEPTR29
COPYFILE23 \DEFINEDEVICE11 \INFER.FILE.TYPE24
DIRECTORYNAME20 \DEFPRINT.BY.NAME6 \IS.NOT.RANDACCESSP18
DIRECTORYNAMEP20 \DELETEFILE12 \MAP-OPEN-STREAMS28
EOF24 \DEVICEEVENT12 \NOFILESINFOFN18
FORCEOUTPUT24 \DO.PARAMS.AT.OPEN14 \NOFILESNEXTFILEFN17
GETEOFPTR25 \EOF.ACTION29 \NULLFILEGENERATOR17
GETFILEINFO25 \EOSERROR29 \OPENFILE13
GETFILEPTR26 \FDEV.DEFPRINT7 \PAGED.REVALIDATEFILE15
GETSTREAMPROP6 \FDEVINSTANCE19 \PAGED.REVALIDATEFILELST15
HOSTNAMEP20 \FILE-CONFLICT17 \PAGED.REVALIDATEFILES15
PUTSTREAMPROP6 \FILE.NOT.OPEN18 \PEEKBIN29
SETFILEINFO26 \FILE.WONT.OPEN18 \PRINT-REVALIDATION-RESULT16
SETFILEPTR26 \FILETYPE.FROM.TYPE26 \REMOVEDEVICE12
STREAM6 \FIXPIN30 \REMOVEDEVICE.NAMES12
STREAMPROP5 \FIXPOUT30 \RENAMEFILE14
\ADD.CONNECTED.DIR21 \FLUSH.OPEN.STREAMS24 \REVALIDATEFILE14
\BACKBIN21 \GENERATEFILEINFO13 \SETACCESS7
\BACKFILEPTR21 \GENERATEFILES13 \SETCLOSEDFILELENGTH29
\BACKPEEKBIN21 \GENERATENEXTFILE13 \SETOFPTR29
\BIN21 \GENERATENOFILES17 \SETFILEPTR29
\BINS21 \GENERIC.BINS27 \STREAM.DEFPRINT7
\BOUT21 \GENERIC.BOUTS27 \STREAM.NOT.OPEN18
\BOUTEOL30 \GENERIC.CHARSET28 \TRUNCATEFILE17
\BOUTS22 \GENERIC.OPENP27 \TYPE.FROM.FILETYPE25
\BUFFERED.BIN31 \GENERIC.OUTFILEP13

MACRO INDEX

APPENDABLE5 READONLY5 \CONVERT-PATHNAME28 \RECOGNIZE-HACK8
APPENDONLY5 SIZE.FROM.LENGTH30 \DECFILEPTR30 \SIGNEDWIN30
DIRTYABLE5 STREAMOP4 \DEVICE-OPEN-STREAMS28 \SIGNEDWOUT30
FDEVOP8 TestMasked5 \EOF30 \WIN30
OPENED5 WRITEABLE5 \GETFILEPTR30 \WOUT30
OVERWRITABLE5 \BINS30 \INHERITFDEVOP.D19
READABLE5 \BOUTS30 \INHERITFDEVOP.S19

VARIABLE INDEX

EOLCONVENTIONS5 STREAM-AFTER-OPEN-FNS12 \FILEDEVICENAMES18
FILELINELENGTH7 SYSTEMRECLST2,10 \FILEDEVICES18
FILING.TYPES28 \CONNECTED.DIRECTORY19 \STREAM.DEFAULT.MAXBUFFERS7
LOGINHOST/DIR19 \DEVICENAMETODEVICE18

CONSTANT INDEX

ANY.EOLC5 BothBits5 CRLF.EOLC5 NoBits5 WordsPerPage31
AppendBit5 ByteOffsetSize31 LF.EOLC5 OutputBits5 WriteBit5
BitsPerByte31 CR.EOLC5 MaxChar31 ReadBit5 \MAXFILEPTR31

RECORD INDEX

BYTEPTR31 FDEV8 FILEGENOBJ10 NOFILEGENSTATE18 STREAM3

PROPERTY INDEX

BOUT27 FILEIO2

OPTIMIZER INDEX

ACCESS-CHARSET29

{MEDLEY}<sources>FILEIO.;1
