

:: TESTER FOR MULTIPLE STREAMS PER FILE

:: The main functionality to test with multiple streams per file is to
:: make sure that the functions listed below work properly using various
:: devices and all of the combinations of the access modes when trying to
:: open more than one stream on the same file. The tests should also
:: verify valid & distinct streams for each legal open case, verify same
:: data at same locations read from each open stream, verify filepos can
:: be set and is unique for each open stream, verify correct results
:: using 2 or more separate processes on same machine, and 1 or more
:: processes on different machines opening multiple streams on the same,
:: verify correct operation of multiple open streams for both NS and IFS
:: files, verify closing one stream does not close any other open stream
:: on the same file, verify that a file cannot be deleted or renamed
:: with an open stream, and use as many device names as possible to open
streams with.

:: Functions To Be Tested:

:: OPENSTREAM, STREAMP, OPENP,
:: CLOSEF, INFILE, OUTFILE, GETFILEINFO,
:: WHENCLOSE, DELFILE,
:: RENAMEFILE, BIN, BOUT, EOF, P,
:: SETFILEPTR, GETFILEPTR, GETSTREAM

::
:: Created By: Jim Blum
::
:: Creation Date: FEB 11, 1987
::
::
:: Last Update: FEB 16, 1987
::
:: Filed As: {ERIS}<LISPCORE>TEST>MSPF>MSPF.TEST
::
::

(DO-TEST-GROUP "TEST MULTIPLE STREAMS PER FILE"
:BEFORE

(PROGN
 (TEST-SETQ DEVICE-LIST '("{DSK}TESTFILE" "{ER
INYES}<TEST>TESTFILE" "{CORE}TESTFILE"
"{PELE:}<PUBLIC>TESTFILE")) ; list of devices/filenames to test
 (TEST-SETQ ACCESS-MODES '(IL:INPUT IL:OUTPUT

```

IL:BOTH IL:APPEND))
    (TEST-SETQ RESULT T)
)
:AFTER
  (PROGN
    (DOLIST (FILE DEVICE-LIST)
      (DO NIL ((NULL (IL:DELFILE FILE)))) ; DELETE ALL VERSIONS
    )
  )
)

```

(DO-TEST "Verify combinations of open access methods work as they should"

```

  (DOLIST (FILE DEVICE-LIST)
    (DO NIL ((NULL (IL:DELFILE FILE)))) ; DELETE OLD VERSIONS
  )

```

```

  (DOLIST (FILE DEVICE-LIST RESULT)
    (UNWIND-PROTECT
      (PROGN
        (TEST-SETQ STREAM (IL:OPENSTREAM FILE 'IL:OUTPUT))
        (DOLIST (MODE ACCESS-MODES)
          (IF (TEST-SETQ STREAM2
              (IGNORE-ERRORS
                (IL:OPENSTREAM FILE MODE

```

```

'IL:OLD)

```

```

    )
  )
  (PROGN
    (IGNORE-ERRORS (IL:CLOSEF
STREAM))
    (FORMAT *ERROR-OUTPUT* "FILE ~A
OPENED ILLEGALLY IN MODE ~A~%~%" FILE MODE)
    (SETQ RESULT NIL)
  ) ; PROGN
) ; IF

```

```

) ; DOLIST
) ; PROGN
  (PROGN (IL:CLOSEF STREAM) (DELETE-FILE FILE))
) ; UNWIND PROTECT
) ; DOLIST
) ; DO-TEST

```

(DO-TEST "TEST MULTIPLE INPUT STREAMS WORK"

```

  (TEST-SETQ RESULT T)
  (DOLIST (FILE DEVICE-LIST)
    (DO NIL ((NULL (IL:DELFILE FILE)))) ; DELETE OLD VERSIONS
  )

```

```

(DOLIST (FILE DEVICE-LIST RESULT)
  (UNWIND-PROTECT
    (PROGN
      (TEST-SETQ STREAM (IL:OPENSTREAM FILE 'IL:OUTPUT))
      (PRINT "THIS IS A TEST" STREAM)
      (IL:CLOSEF STREAM)
      (TEST-SETQ STREAM (IL:OPENSTREAM FILE 'IL:INPUT))
      (IF (NOT (AND (IL:OPENP STREAM)
                    (IL:GETSTREAM STREAM)
                    (STREAMP STREAM)
                    )
          )
        (PROGN
          (FORMAT *ERROR-OUTPUT* "FILE ERROR IN ~A~%~%" FILE)
          (TEST-SETQ RESULT NIL)
        )
      )
    )
  (TEST-SETQ STREAM2 (IL:OPENSTREAM FILE 'IL:INPUT))
  (IF (NOT (AND (IL:OPENP STREAM2)
                (IL:GETSTREAM STREAM2)
                (STREAMP STREAM2)
                )
      )
    (PROGN (FORMAT *ERROR-OUTPUT* "Cannot open multiple
stream on file ~a~%~%" FILE)
      (TEST-SETQ RESULT NIL)
    )
  )
  (TEST-SETQ STREAM3 (IL:OPENSTREAM FILE 'IL:INPUT))
  (IF (NOT (AND (IL:OPENP STREAM3)
                (IL:GETSTREAM STREAM3)
                (STREAMP STREAM3)
                )
      )
    (PROGN
      (FORMAT *ERROR-OUTPUT* "Cannot open multiple
stream on file ~a~%~%" FILE)
      (TEST-SETQ RESULT NIL)
    )
  )
  (IL:SETFILEPTR STREAM 0)
  (IL:SETFILEPTR STREAM2 0)
  (IL:SETFILEPTR STREAM3 0)
  (IF (NOT (EQUAL (READ STREAM)
                  "THIS IS A TEST"
                  )
      )
  )

```

```

    )
    (TEST-SETQ RESULT NIL)
  )
  (IF (NOT (EQUAL (READ STREAM2)
                  "THIS IS A TEST"
                  )
    )
    )
    (TEST-SETQ RESULT NIL)
  )
  (IF (NOT (EQUAL (READ STREAM3)
                  "THIS IS A TEST"
                  )
    )
    )
    (TEST-SETQ RESULT NIL)
  )
  (IF (OR (IL:NEQ (IL:GETFILEPTR STREAM) 17)
          (IL:NEQ (IL:GETFILEPTR STREAM2) 17)
          (IL:NEQ (IL:GETFILEPTR STREAM3) 17)
        )
    )
    (PROGN
      (FORMAT *ERROR-OUTPUT* "FILEPTR ON STREAM IS
NOT CORRECT~%~%")
      (TEST-SETQ RESULT NIL)
    )
  )
  (IF (OR (IL:EOFP STREAM)
          (IL:EOFP STREAM2)
          (IL:EOFP STREAM3)
        )
    )
    (PROGN
      (FORMAT *ERROR-OUTPUT* "EARLY END OF FILE
ENCOUNTERED~%~%")
      (TEST-SETQ RESULT NIL)
    )
  )
  )
  (IGNORE-ERRORS (READ STREAM))
  (IGNORE-ERRORS (READ STREAM2))
  (IGNORE-ERRORS (READ STREAM3))
  (IF (NOT (AND (IL:EOFP STREAM)
                (IL:EOFP STREAM2)
                (IL:EOFP STREAM3)
              )
    )
    )
    (PROGN (FORMAT *ERROR-OUTPUT* "END OF FILE NOT
ENCOUNTERED~%~%")
      (TEST-SETQ RESULT NIL)
    )
  )

```

```

    ) ; IF
    (IF (IGNORE-ERRORS (IL:RENAMEFILE FILE 'TESTFILE2))
        (PROGN
          (FORMAT *ERROR-OUTPUT* "RENAME-FILE
ERROR~%~%"
          (TEST-SETQ RESULT NIL)
        )
    )
    (IF (IGNORE-ERRORS (IL:DELFILE FILE))
        (PROGN
          (FORMAT *ERROR-OUTPUT* "FILE ~A DELETED WHILE A
STREAM WAS OPEN ON IT~%~%" FILE)
          (TEST-SETQ RESULT NIL)
        )
    )
    ) ; PROGN FOR UNWIND-PROTECT
    (PROGN (IGNORE-ERRORS (IL:CLOSEF STREAM))
            (IGNORE-ERRORS (IL:CLOSEF STREAM2))
            (IGNORE-ERRORS (IL:CLOSEF STREAM3))
            (IF (OR (IL:OPENP STREAM)
                    (IL:OPENP STREAM2)
                    (IL:OPENP STREAM3)
                )
                (PROGN (FORMAT *ERROR-OUTPUT*
"IL:OPENP RETURNS NON-NIL ON A CLOSED STREAM~%~%"
                        (TEST-SETQ RESULT NIL)
                )
            )
    ) ; IF
    (IGNORE-ERRORS (DELETE-FILE FILE))
  ) ; PROGN
) ; UNWIND-PROTECT
) ; DOLIST
) ; DO-TEST

```

```

(DO-TEST "TEST FILE ATTRIBUTES UNAFFECTED WITH MULTIPLE
STREAMS"
  (TEST-SETQ RESULT T)
  (DOLIST (FILE DEVICE-LIST)
    (DO NIL ((NULL (IL:DELFILE FILE)))) ; DELETE OLD VERSIONS
  )
  (DOLIST (FILE DEVICE-LIST RESULT)
    (UNWIND-PROTECT
      (PROGN
        (TEST-SETQ STREAM (IL:OPENSTREAM FILE 'IL:OUTPUT))
      )
    )
  )

```



```

                                                    (IL:PUSH CLIST
(READ ST))))
  (IL:WHENCLOSE STREAM2 'IL:BEFORE (FUNCTION (LAMBDA
(ST)
                                                    (IL:PUSH CLIST
(READ ST))))
  (IL:WHENCLOSE STREAM3 'IL:BEFORE (FUNCTION (LAMBDA
(ST)
                                                    (IL:PUSH CLIST
(READ ST))))
  )
  (SETQ RESULT
    (PROGN
      (IGNORE-ERRORS (IL:CLOSEF STREAM))
      (IGNORE-ERRORS (IL:CLOSEF STREAM2))
      (IGNORE-ERRORS (IL:CLOSEF STREAM3))
      (EQUAL CLIST '("THIS IS A TEST" "THIS IS A TEST" "THIS IS A
TEST"))
    ) ; PROGN
  ) ; SETQ
  ) ; UNWIND-PROTECT
  ) ; DOLIST
  ) ; DO-TEST

(DO-TEST "TEST INFILE & OUTFILE"
  (TEST-SETQ RESULT NIL)
  (DOLIST (FILE DEVICE-LIST RESULT)
    (UNWIND-PROTECT
      (PROGN
        (IL:OUTFILE FILE)
        (IL:PRINT "THIS IS A TEST" NIL)
        (IL:CLOSEF NIL)
        (SETQ STREAM (IL:INFILE FILE))
        (SETQ STREAM2 (IL:OPENSTREAM FILE 'IL:INPUT))
      )
    )
  (SETQ RESULT
    (AND
      (EQUAL (IL:READ NIL) "THIS IS A TEST")
      (NOT (IGNORE-ERRORS (IL:CLOSEF STREAM)))
      (IGNORE-ERRORS (IL:CLOSEF STREAM2))
      (IGNORE-ERRORS (IL:CLOSEF NIL))
      (IL:TRUE (IL:CLOSEF NIL))
    )
  ) ; PROGN
  ) ; SETQ
  ) ; UNWIND-PROTECT

```

```
) ; DOLIST  
) ; DO-TEST  
) ; DO-TEST-GROUP
```

```
STOP
```