```
; rw-email.lisp
; Created on March 11, 2013 by Wesley R. Howell.
; Purpose:
; This file will contain the dependencies to other .lisp files and
; handled the IO portion of the client functions.
; CHANGE LOG:
; 4/ 1/2013 - Changed the file path for outputted files to be consistant with
            new Java interface.
; 3/11/2013 - Created file to handled the client logic for email messages
; ------
(in-package "ACL2")
(include-book "../../include/xml-scanner" :uncertified-okp t)
(include-book "../../include/io-utilities" :uncertified-okp t)
(include-book "../client-email" :uncertified-okp t)
(set-state-ok t)
;(readEmailText file state)
; Reads an email message, then outputs the contenets to the console.
;Working on getting it to write out to an HTML file.
(defun readEmailText (file state)
  (mv-let (input-xml error-open state)
         (file->string file state)
    (if error-open
        (mv error-open state)
        (mv (getEmailText (cdr (cdr (tokenizeXML input-xml)))) state))))
(defun readEmailHTML (file state)
  (mv-let (input-xml error-open state)
         (file->string file state)
    (if error-open
        (mv error-open state)
        (mv (getEmailHTML (cdr (cdr (tokenizeXML input-xml)))) state))))
(defun readEmail (fin fout state)
  (mv-let (input-as-string error-open state)
     (file->string fin state)
    (if error-open
        (mv error-open state)
        (mv-let (error-close state)
               (string-list->file
                (concatenate 'string "store/email/inbox/"
                            "msg_"
                            fout
                            ".html")
                              (getEmailHTML
                                     (cdr (cdr
                                      (tokenizeXML input-as-string)))
                                     ;These two cdr's remove the two XML
                                     ;headers for the file
                               )
                                state)
           (if error-close
              (mv error-close state)
              (mv "Success File has been written!"
                   state))))))
(defun writeEmailToFile (xmlStr to ts state)
  (mv-let (error state)
         (string-list->file (concatenate 'string
                                      "store/email/outbox/"
```