```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; route-email.lisp
;
; Created on February 28, 2013 by Wesley R. Howell.
;
; Purpose:
; This file will execute the email transition between clients. In order
; for this to happen, the incoming email will need to be opened, then the
; to field will need to be parsed into a list. Once this list is generated
; we can then send a copy of the email message to the receipent clients.
;
; CHANGE LOG:
; 3/4/2013  Added 5 lines to the RWEmail function to take in a timestamp
;           and dynamically write the email to the appropriate directory
;           based on the <to> tag in the email message.
; 2/28/2013 Created this file to handled the IO and outside dependencies
;           of the server-email functions.
; ---------------------------------------------------------------------
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(in-package "ACL2")

;;Outside dependencies for this module
(include-book "../../../include/xml-scanner" :uncertified-okp t)
(include-book "../../../include/io-utilities" :uncertified-okp t)
(include-book "../server-email" :uncertified-okp t)
(set-state-ok t)

;;(writeEmail email fOut state)
;;Writes an email message based on the email data structure with
;;a single email address. Calls the getEmailXML function to generate
;;the output text.
;;email - the email data structure to output
;;fOut - the file path for the outputted file
;;state - the ACL2 state
(defun writeEmail (email fOut state)
  (mv-let (error-close state)
          (string-list->file
           fOut
           (getEmailXML email)
           state
           )
          (if error-close
              (mv error-close state)
              (mv (string-append ", output file: " fOut)
                  state)
           )))


;;;getEmailXMLTokens file
;;;This function will return the list of XML tokens for the email XML
;;;file - the filepath to the XML file
(defun getEmailXMLTokens (file state)
  (mv-let (input-xml error-open state)
          (file->string file state)
     (if error-open
         (mv error-open state)
         (mv (cdr (cdr (tokenizeXML input-xml))) state))))

;(rwEmail fin fout state)
;This function is the entry point to parse a single email message from the
;XML file "fin"
;fin - input XML file
;fout - output XML filename
;state - ACL2 state
(defun rwEmail (#|fin|#input-as-string 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/"
                               (car (cdr (getContactStructure (car(car (cddddr (tokenizeXML input-as-       ↙
    string)))))))
                                                              "/"
                               (car (getContactStructure (car(car (cddddr (tokenizeXML input-as-           ↙
    string))))))
                   ;Separate this into the structure ../domain/user
                               "/msg_"
                               fout
                               ".xml")
                                  (getEmail
                                         (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))));))
```