```
;
; Created on March 24, 2013 by Adam Ghodratnama
;
; Purpose:
; This file contains the functions that are essential to creating a request
; from the client side to block a user from sending emails to a client
;
; CHANGE LOG:
; -------------------------------------------------------------------------
; 2013-03-25    -   Initial file conception.
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(in-package "ACL2")
(include-book "../../../include/io-utilities")
(include-book "../../../include/xml-scanner")

(set-state-ok t)
(set-ignore-ok t)

; (addressXML address)
; Generates the XML form for an address from the address structure minus.
; the password.
; address - the address structure in the form of (domain name).
(defun addressXML (address)
    (if (endp address)
        nil
            ; Domain tag <domain>#PCDATA</domain>
        (let* ((domain (concatenate 'string
                           "<domain>" (car address) "</domain>"))
                       ; Name tag <name>#PCDATA</name>
                       (name (concatenate 'string
                           "<name>" (cadr address) "</name>"))
                       ; Create a list with the address fields in it
                       (regRequest (list domain name)))
            regRequest)))

;
; The purpose of this function is to create an XML file that requests
; that a user is blocked from emailing the client for a user.
;
; The function will create an XML file containing the two addresses:
; 1. The address of the blocker
; 2. The address of the user to be blocked
;
; When this structure is sent to the server it will allow identification
; of both parties so that the appropriate actions are taken on the
; respective parties.
;
; Example call:
; (createRequest '("localhost" "blocker")
;     '("localhost" "blockee") timeStamp state)
(defun createRequest (blockerAddress blockeeAddress timeStamp state)
  (let* ((blockerDomain (car (addressXML blockerAddress)))
         (blockerName (cadr (addressXML blockerAddress)))
         (blockeeDomain (car (addressXML blockeeAddress)))
         (blockeeName (cadr (addressXML blockeeAddress))))
    (mv-let (error state)
            (string-list->file
             (concatenate 'string
                  (string-append "../../../store/user/requests/block/request-block" (rat->str timeStamp 0))
                  ".xml")
             (list
                  "<?xml version='1.0'?>"
                  ;"<!DOCTYPE user SYSTEM '../../../dtd/register-user.dtd'>" ; NEED DTD FOR BLOCK USER
                  "<blocker>"
                  blockerDomain
                  blockerName
```

```
      "</blocker>"
      "<blockee>"
      blockeeDomain
      blockeeName
      "</blockee>")
 state)
(if error
    (mv error state)
    (mv "Wrote block user request" state)))))
```