```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
; register-user.lisp
;
; Created on February 14, 2013 by Matthew A. Crist.
;
; Purpose:
; This file contains the functions that are essential to adding a user to
; the address book.
;
; CHANGE LOG:
; -----------------------------------------------------------------------
; 2013-03-17    -   Added uncertified book ignore on load.
; 2013-03-17    -   Added the password to be added to address-book on
;                 registration.
; 2013-02-14    -   Initial conception of this file.
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

(in-package "ACL2")

(include-book "../../../include/io-utilities" :uncertified-okp t)
(include-book "../../../include/xml-scanner" :uncertified-okp t)
(include-book "../address-book" :uncertified-okp t)

; (getDomain tokens)
; Acquires the domain of the registration request from the token list.
; tokens    -   the scanned tokens from the incoming XML register request.
(defun getDomain (tokens)
    (if (endp tokens)
        nil
        (if (equal "<domain>" (caar tokens))
            (caadr tokens)
            (getDomain (cdr tokens)))))

; (getName tokens)
; Acquires the name of the registration request from the token list.
; tokens    -   the scanned tokens from the incoming XML register request.
(defun getName (tokens)
    (if (endp tokens)
        nil
        (if (equal "<name>" (caar tokens))
            (caadr tokens)
            (getName (cdr tokens)))))

; (getPassword tokens)
; Acquires the password of the registration request from the token list.
; tokens    -   the scanned tokens from the incoming XML register request.
(defun getPassword (tokens)
    (if (endp tokens)
        nil
        (if (equal "<password>" (caar tokens))
            (caadr tokens)
            (getPassword (cdr tokens)))))

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

; (registerUser regXML abXML state)
; Processes the information that is passed via XML string to add the user
; to the global server address book.
; regXML - The XML that is contained in the registration file - sent
;          dynamically via shell script.
; abXML  - The XML that is contained in the address book file - sent
;          dynamically via shell script.
; state  - The state of the streams in ACL2.
(defun registerUser (regXML abXML state)
    (let* ((tokens (tokenizeXML regXML))
             (domain (getDomain tokens))
             (name (getName tokens))
```

```
     (pass (getPassword tokens))
     (addressBook (getAddressBook (tokenizeXML abXML))))
(mv-let (error state)
    (string-list->file
       "store/address-book/temp_address-book.xml"
       (getAddressBookXML (addAddress addressBook (list domain name pass)))
        state)
    (if error
        (mv error state)
        (mv "Wrote temp_address-book.xml successfully" state)))))
```