

```
; Created on March 7, 2013 by Adam Ghodrathnama
;
; Purpose:
; This file contains the functions that are essential to creating a request
; from the client side to register a user on the server.
;
; CHANGE LOG:
; -----
; March 10, 2013 - Completion of createRequest function
;
; March 17, 2013 - Changed parameters of createRequests function to accept
;                  an argument for the time stamp of creation used by the
;                  shell script
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
(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.
; address - the address structure in the form of (domain name password).
(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>"))
              ; Password tag <password>#PCDATA</password>
              (password (concatenate 'string
                                      "<password>" (caddr address) "</password>"))
              ; Create a list with the address fields in it
              (regRequest (list domain name password)))
            regRequest)))

;
; The purpose of this function is to create an XML file that requests
; access for a user.
;
; The function will create an XML file containing the domain, name and
; password for a user
;
; Example call: (createRequest '("localhost" "adam" "password") timeStamp state)
;
(defun createRequest (address timeStamp state)
  (let* ((domain (car (addressXML address)))
         (name (cadr (addressXML address)))
         (pass (caddr (addressXML address))))
    (mv-let (error state)
      (string-list->file
       (concatenate 'string
                     (string-append "store/user/requests/register/request-register" (rat->str timeStamp 0))
                     ".xml")
       (list
        "<?xml version='1.0'?>"
        "<!DOCTYPE user SYSTEM '.././../dtd/register-user.dtd'>"
        "<user>"
        domain
        name
        pass
        "</user>")
       state)
      state)
```

```
(if error
  (mv error state)
  (mv "Wrote registration request" state))))
```